8. Least squares

- Review of linear equations
- Least squares
- Example: curve-fitting
- Vector norms
- Geometrical intuition
- Example: moving average model

Review of linear equations

System of m linear equations in n unknowns:

$$\begin{aligned} a_{11}x_1 + \cdots + a_{1n}x_n &= b_1 \\ a_{21}x_1 + \cdots + a_{2n}x_n &= b_2 \\ \vdots \quad & \quad \vdots \quad \quad \vdots \\ a_{m1}x_1 + \cdots + a_{mn}x_n &= b_m \end{aligned} \iff \begin{bmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{m1} & \cdots & a_{mn} \end{bmatrix} \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ \vdots \\ b_m \end{bmatrix}$$

Compact representation: $Ax = b$. Only three possibilities:

1. exactly one solution (e.g. $x_1 + x_2 = 3$ and $x_1 - x_2 = 1$)

2. infinitely many solutions (e.g. $x_1 + x_2 = 0$)

3. no solutions (e.g. $x_1 + x_2 = 1$ and $x_1 + x_2 = 2$)

Review of linear equations

- **column interpretation**: the vector b is a linear combination of $\{a_1, \ldots, a_n\}$, the columns of A.

$$Ax = \begin{bmatrix} a_1 & a_2 & \ldots & a_n \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = a_1 x_1 + \cdots + a_n x_n = b$$

  The solution x tells us how the vectors a_i can be combined in order to produce b.

- can be visualized in the output space $\mathbb{R}^m$.

Review of linear equations

- **row interpretation**: the intersection of hyperplanes $\tilde{a}_i^T x = b_i$ where $\tilde{a}_i^T$ is the i^{th} row of A.

$$Ax = \begin{bmatrix} \tilde{a}_1^T \\ \tilde{a}_2^T \\ \vdots \\ \tilde{a}_m^T \end{bmatrix} x = \begin{bmatrix} \tilde{a}_1^T x \\ \tilde{a}_2^T x \\ \vdots \\ \tilde{a}_m^T x \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{bmatrix}$$

  The solution x is a point at the intersection of the affine hyperplanes. Each $\tilde{a}_i$ is a normal vector to a hyperplane.

- can be visualized in the input space $\mathbb{R}^n$.

Review of linear equations

- The set of solutions of $Ax = b$ is an **affine subspace**.
- If $m > n$, there is (usually but not always) no solution. This is the case where A is **tall** (overdetermined).
  - Can we find x so that $Ax \approx b$?
  - One possibility is to use **least squares**.
- If $m < n$, there are infinitely many solutions. This is the case where A is **wide** (underdetermined).
  - Among all solutions to $Ax = b$, which one should we pick?
  - One possibility is to use **regularization**.

In this lecture, we will discuss **least squares**.

Least squares

- Typical case of interest: $m > n$ (overdetermined). If there is no solution to $Ax = b$, we try instead to make $Ax \approx b$.
- The least-squares approach: make Euclidean norm $\|Ax - b\|$ as small as possible.
- Equivalently: make $\|Ax - b\|^2$ as small as possible.

Standard form:

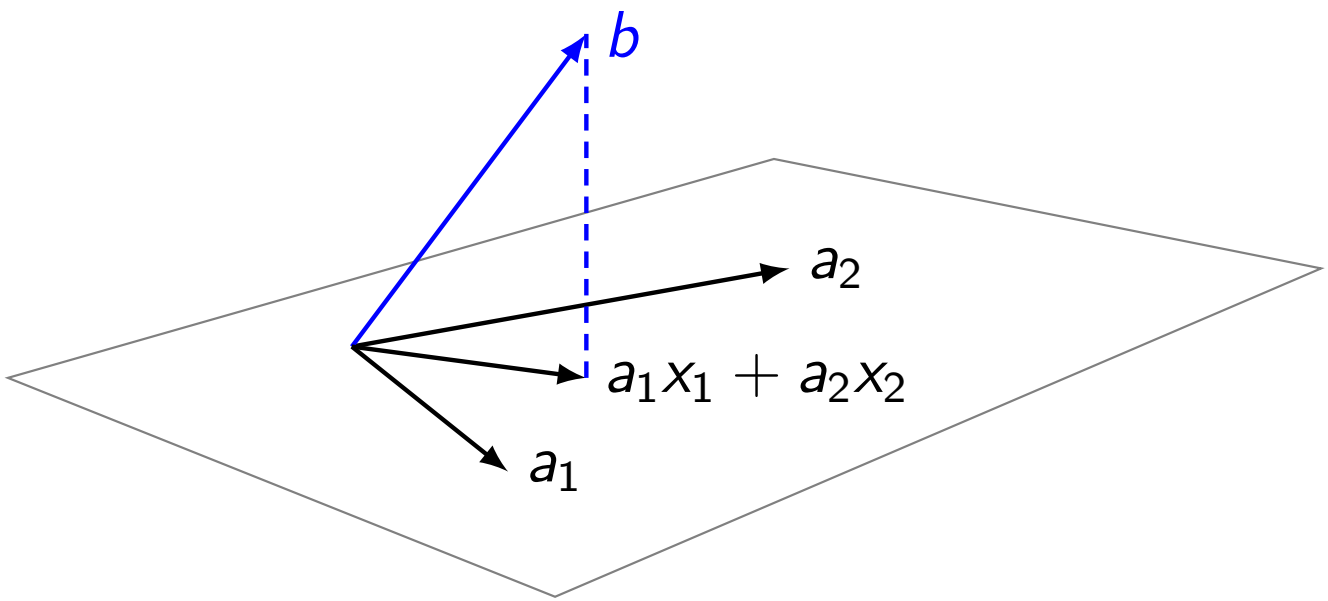
$$\underset{x}{\text{minimize}} \quad \|Ax - b\|^2$$

It's an unconstrained optimization problem.

Least squares

- **column interpretation**: find the linear combination of columns $\{a_1, \ldots, a_n\}$ that is closest to b.

$$\|Ax - b\|^2 = \left\|(a_1x_1 + \cdots + a_nx_n) - b\right\|^2$$

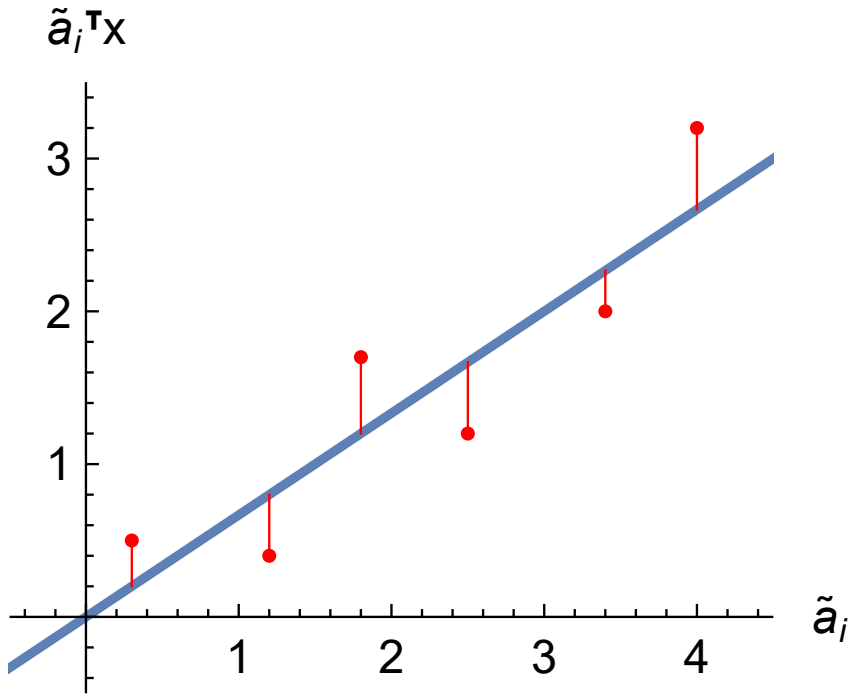


Least squares

- **row interpretation**: If $\tilde{a}_i^T$ is the i^{th} row of A,

$$\|Ax - b\|^2 = (\tilde{a}_1^T x - b_1)^2 + \cdots + (\tilde{a}_m^T x - b_m)^2$$

Here, x is the slope of the blue line and the red dots are b_i.
Goal: minimize sum of squares of residuals (vertical lines).



JuMP implementation

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

When defining your objective in JuMP, use one of:

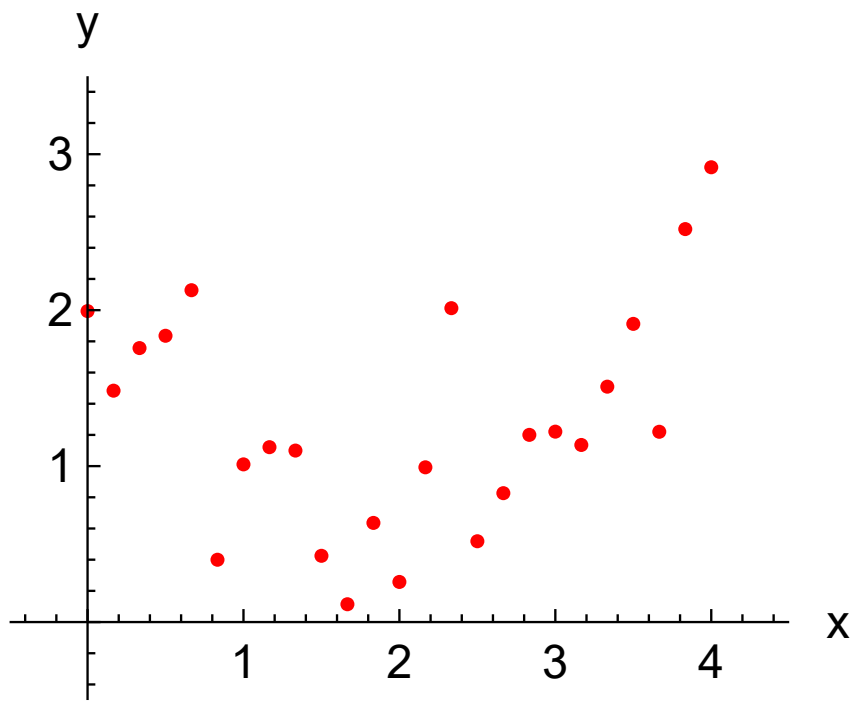
- `sum((A*x-b).^2)`
- `(A*x-b)'*(A*x-b)`
- `dot(A*x-b,A*x-b)` (requires `using LinearAlgebra`)

Do not use `norm(x)` or `norm(x)^2`.

Valid solvers: HiGHS, DAQP, OSQP, ECOS, Gurobi, Mosek

Example: curve-fitting

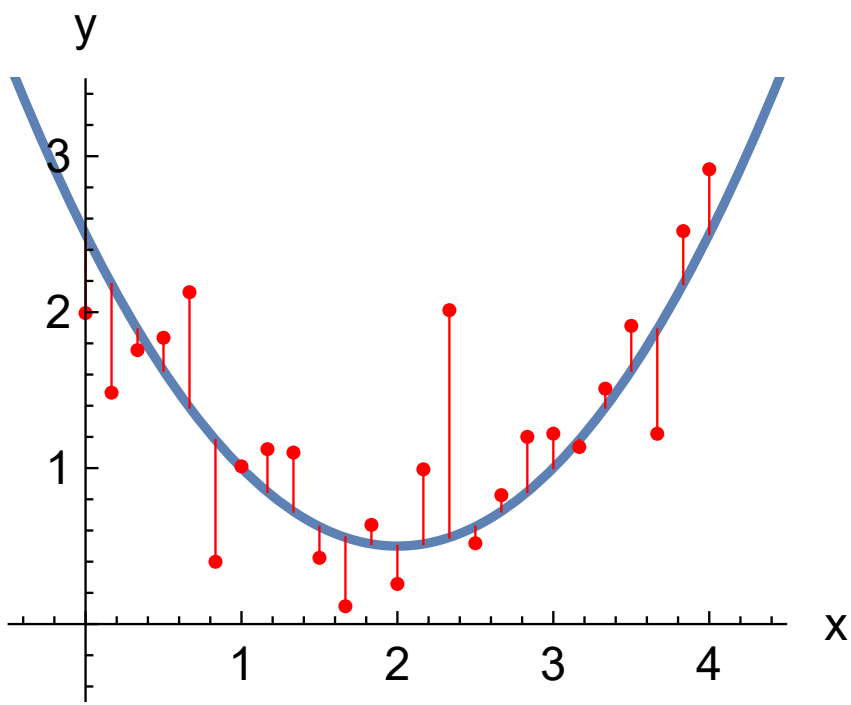
- We are given noisy data points (x_i, y_i).
- We suspect they are related by $y = px^2 + qx + r$
- Find the p, q, r that best agrees with the data.



Example: curve-fitting

- We want $y_i \approx px_i^2 + qx_i + r$ for all i.
- Minimize the sum of squares of residuals!

$$\underset{p,q,r}{\text{minimize}} \quad \sum_{i=1}^{n} \left(px_i^2 + qx_i + r - y_i\right)^2$$



Example: curve-fitting

Writing all the equations:

$$\begin{aligned} y_1 &\approx px_1^2 + qx_1 + r \\ y_2 &\approx px_2^2 + qx_2 + r \\ &\vdots \\ y_m &\approx px_m^2 + qx_m + r \end{aligned} \quad \Longrightarrow \quad \underbrace{\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_m \end{bmatrix}}_{y} \approx \underbrace{\begin{bmatrix} x_1^2 & x_1 & 1 \\ x_2^2 & x_2 & 1 \\ \vdots & \vdots & \vdots \\ x_m^2 & x_m & 1 \end{bmatrix}}_{A} \underbrace{\begin{bmatrix} p \\ q \\ r \end{bmatrix}}_{u}$$

$$\underset{u}{\text{minimize}} \quad \|Au - y\|^2$$

- Also called **linear regression**

Example: curve-fitting

- **More complicated**: $y = pe^x + q\cos(x) - r\sqrt{x} + s\,x^3$
- Find the p, q, r, s that best agrees with the data.

Writing all the equations:

$$\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_m \end{bmatrix} \approx \begin{bmatrix} e^{x_1} & \cos(x_1) & -\sqrt{x_1} & x_1^3 \\ e^{x_2} & \cos(x_2) & -\sqrt{x_2} & x_2^3 \\ \vdots & \vdots & \vdots & \vdots \\ e^{x_m} & \cos(x_m) & -\sqrt{x_m} & x_m^3 \end{bmatrix} \begin{bmatrix} p \\ q \\ r \\ s \end{bmatrix}$$

- Julia notebook: Regression.ipynb

Vector norms

We want to solve $Ax = b$, but there is no solution. Define the **residual** to be the quantity $r := Ax - b$. We can't make it zero, so instead we try to make it *small*. Many options!

- minimize the largest component (a.k.a. the ∞-norm)

$$\|r\|_\infty = \max_i |r_i|$$

- minimize the sum of absolute values (a.k.a. the 1-norm)

$$\|r\|_1 = |r_1| + |r_2| + \cdots + |r_m|$$

- minimize the Euclidean norm (a.k.a. the 2-norm)

$$\|r\|_2 = \|r\| = \sqrt{r_1^2 + r_2^2 + \cdots + r_m^2}$$

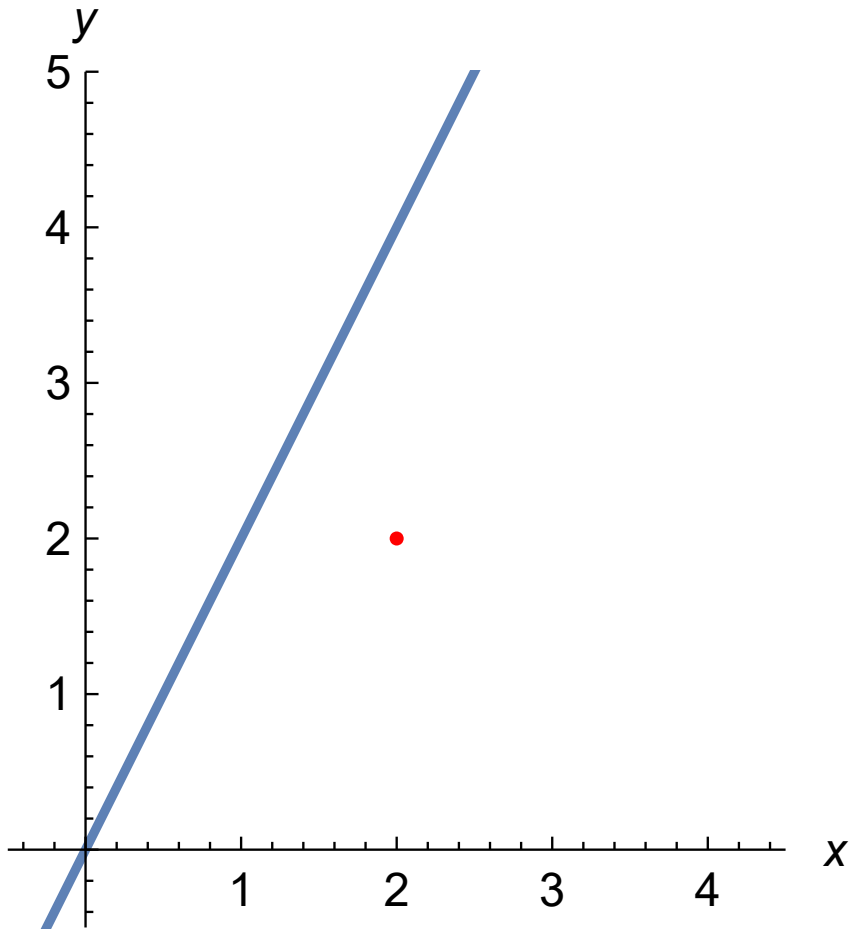
Vector norms

Example: find $\begin{bmatrix} x \\ 2x \end{bmatrix}$ that is closest to $\begin{bmatrix} 2 \\ 2 \end{bmatrix}$.

Blue line is the set of points with coordinates $(x, 2x)$.

Find the one closest to the red point located at $(2, 2)$.

Answer depends on your notion of distance!



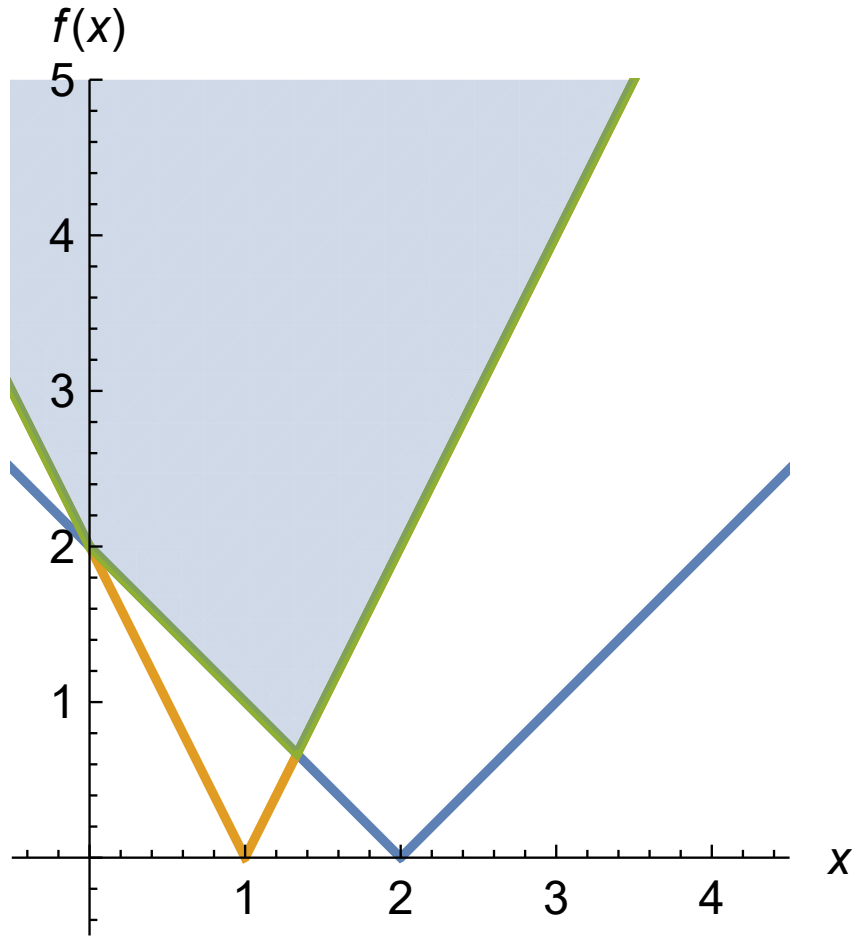
Vector norms

Example: find $\begin{bmatrix} x \\ 2x \end{bmatrix}$ that is closest to $\begin{bmatrix} 2 \\ 2 \end{bmatrix}$.

Minimize largest component:

$$\min_x \max\{|x-2|, |2x-2|\}$$

Optimum is at $x \approx 1.33333$.



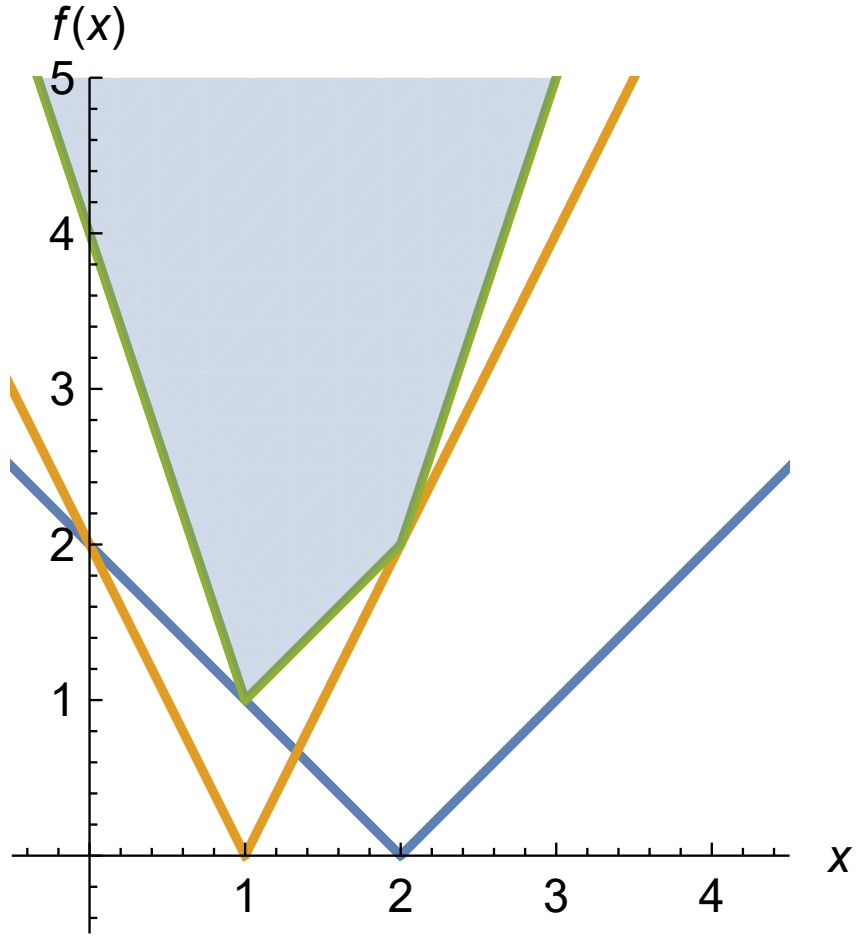
Vector norms

Example: find $\begin{bmatrix} x \\ 2x \end{bmatrix}$ that is closest to $\begin{bmatrix} 2 \\ 2 \end{bmatrix}$.

Minimize sum of components:

$$\min_x \; |x - 2| + |2x - 2|$$

Optimum is at $x = 1$.



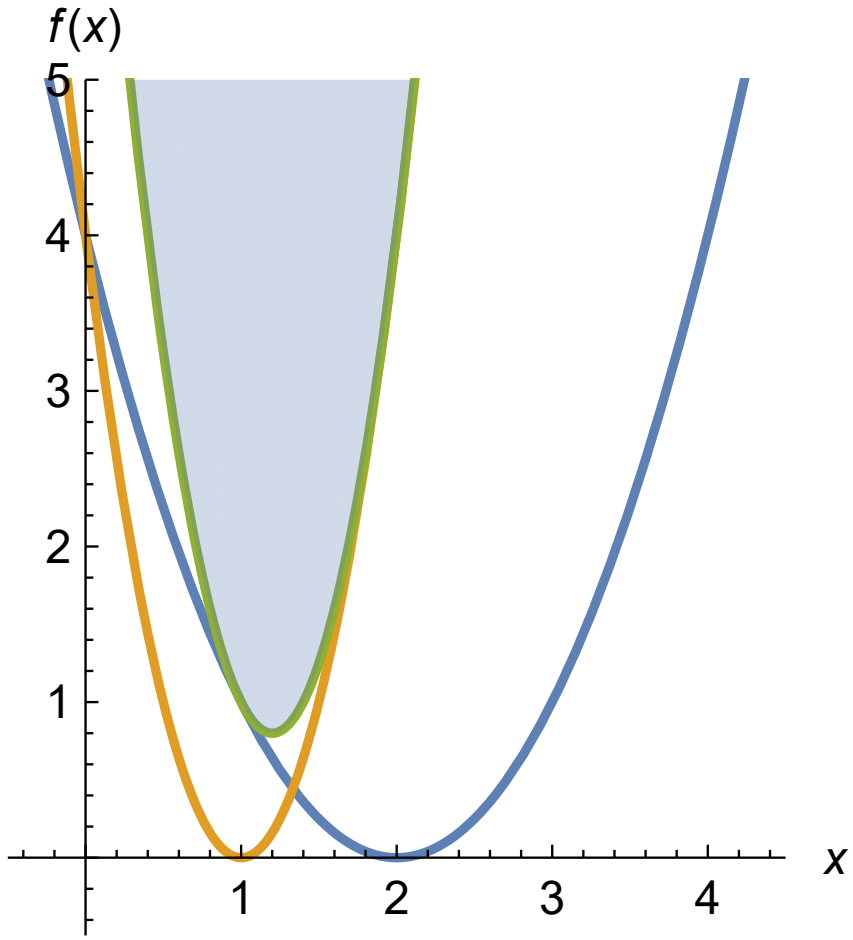
Vector norms

Example: find $\begin{bmatrix} x \\ 2x \end{bmatrix}$ that is closest to $\begin{bmatrix} 2 \\ 2 \end{bmatrix}$.

Minimize sum of squares:

$$\min_x (x-2)^2 + (2x-2)^2$$

Optimum is at $x = 1.2$.



Vector norms

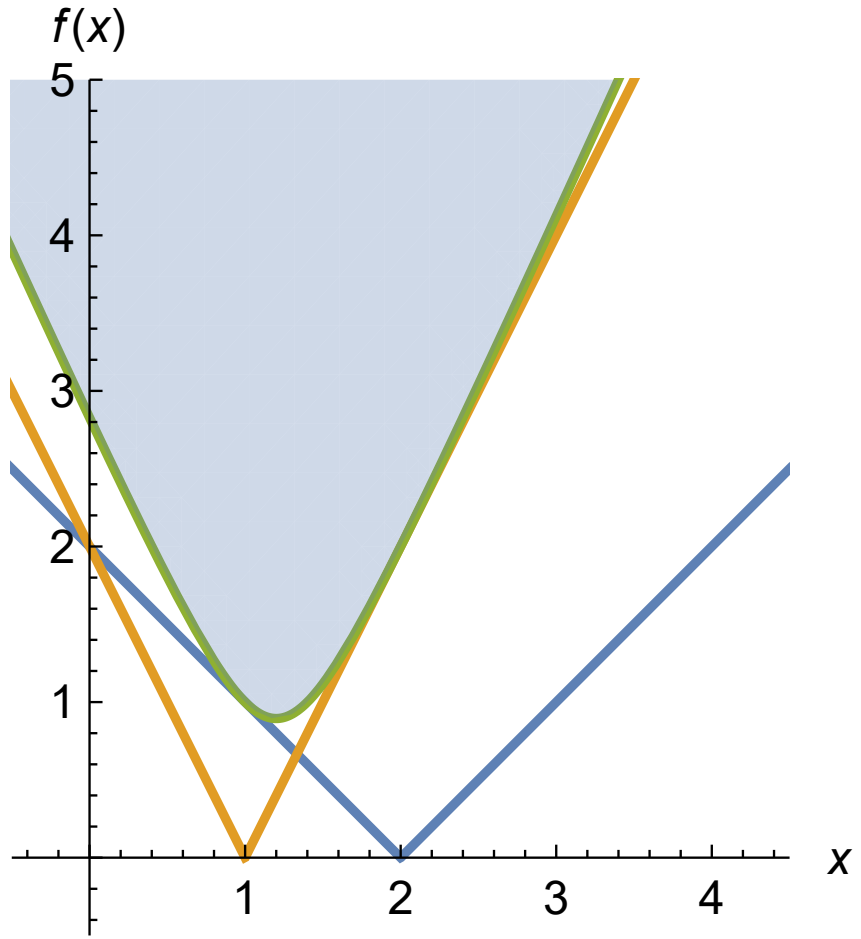
Example: find $\begin{bmatrix} x \\ x \end{bmatrix}$ that is closest to $\begin{bmatrix} 1 \\ 2 \end{bmatrix}$.

Equivalently, we can:

Minimize $\sqrt{\text{sum of squares}}$

$$\min_x \sqrt{(x-2)^2 + (2x-2)^2}$$

Optimum is at $x = 1.2$.



Vector norms

- minimizing the largest component is an LP:

$$\min_x \ \max_i \left|\tilde{a}_i^T x - r_i\right| \quad \Longleftrightarrow \quad \begin{array}{ll} \min\limits_{x,t} & t \\ \text{s.t.} & -t \leq \tilde{a}_i^T x - r_i \leq t \end{array}$$

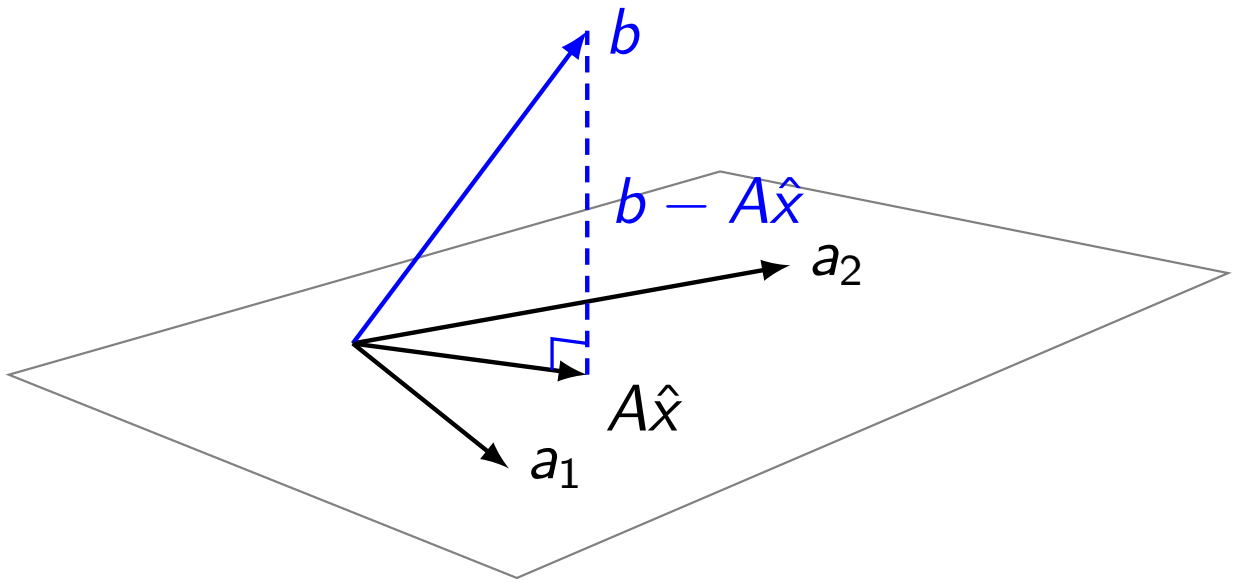
- minimizing the sum of absolute values is an LP:

$$\min_x \ \sum_{i=1}^m \left|\tilde{a}_i^T x - r_i\right| \quad \Longleftrightarrow \quad \begin{array}{ll} \min\limits_{x,t_i} & t_1 + \cdots + t_m \\ \text{s.t.} & -t_i \leq \tilde{a}_i^T x - r_i \leq t_i \end{array}$$

- minimizing the 2-norm is not an LP!

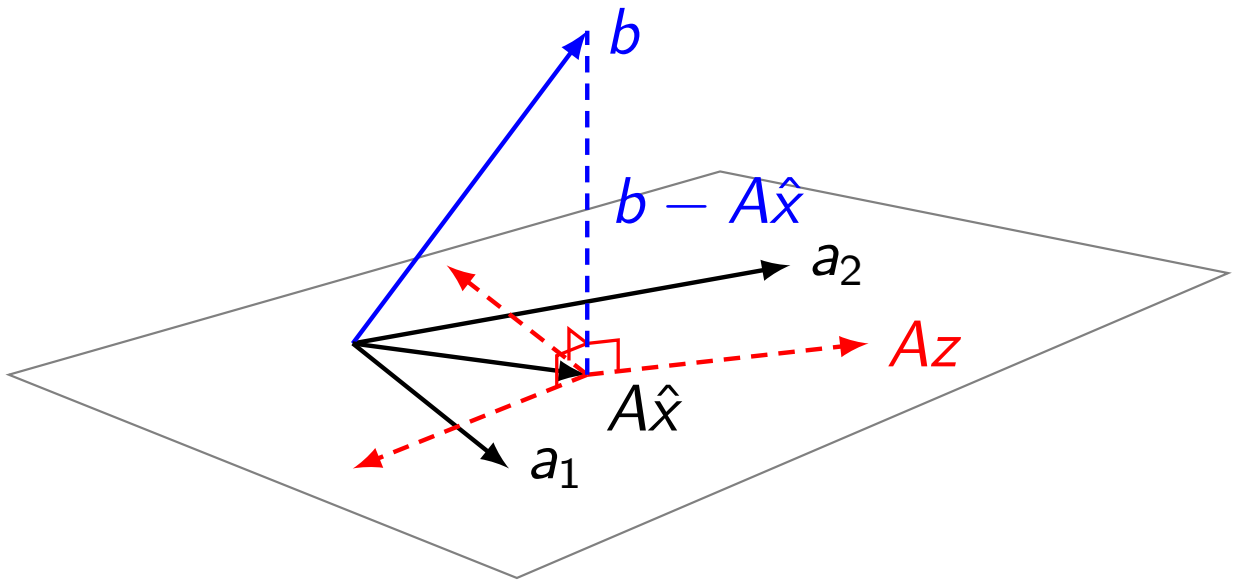
$$\min_x \ \sum_{i=1}^m \left(\tilde{a}_i^T x - r_i\right)^2$$

Geometry of LS



- The set of points $\{Ax\}$ forms a **subspace** (the *range* of A). The vector b does not belong to this subspace.
- If $A\hat{x}$ is as close as possible to b, then the residual $(b - A\hat{x})$ must be orthogonal to the vector $A\hat{x}$.
- This is not enough to guarantee optimality!

Geometry of LS



- Must have: $(A\hat{x} - Az)^{\mathsf{T}}(b - A\hat{x}) = 0$ for all z
- Simplifies to: $(\hat{x} - z)^{\mathsf{T}}(A^{\mathsf{T}}b - A^{\mathsf{T}}A\hat{x}) = 0$. Since this holds for all z, the **normal equations** are satisfied:

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

Optimality conditions

Theorem: The following statements are equivalent:

1. $\hat{x}$ is a solution of the least squares problem

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

2. $\hat{x}$ is a solution to the normal equations

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

- The normal equations always have at least one solution.
- The LS problem has infinitely many solutions precisely when the normal equations have infinitely many solutions.

Normal equations

Least squares problems are **easy** to solve!

- Solving a least squares problem is the same as solving the normal equations.
- 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.

Least squares in Julia

1. Using JuMP, as before:

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

2. Solving the normal equations directly:

   `x = inv(A'*A)*(A'*b)`

   Note: Requires A to have full column rank ($A^\mathsf{T}A$ invertible)

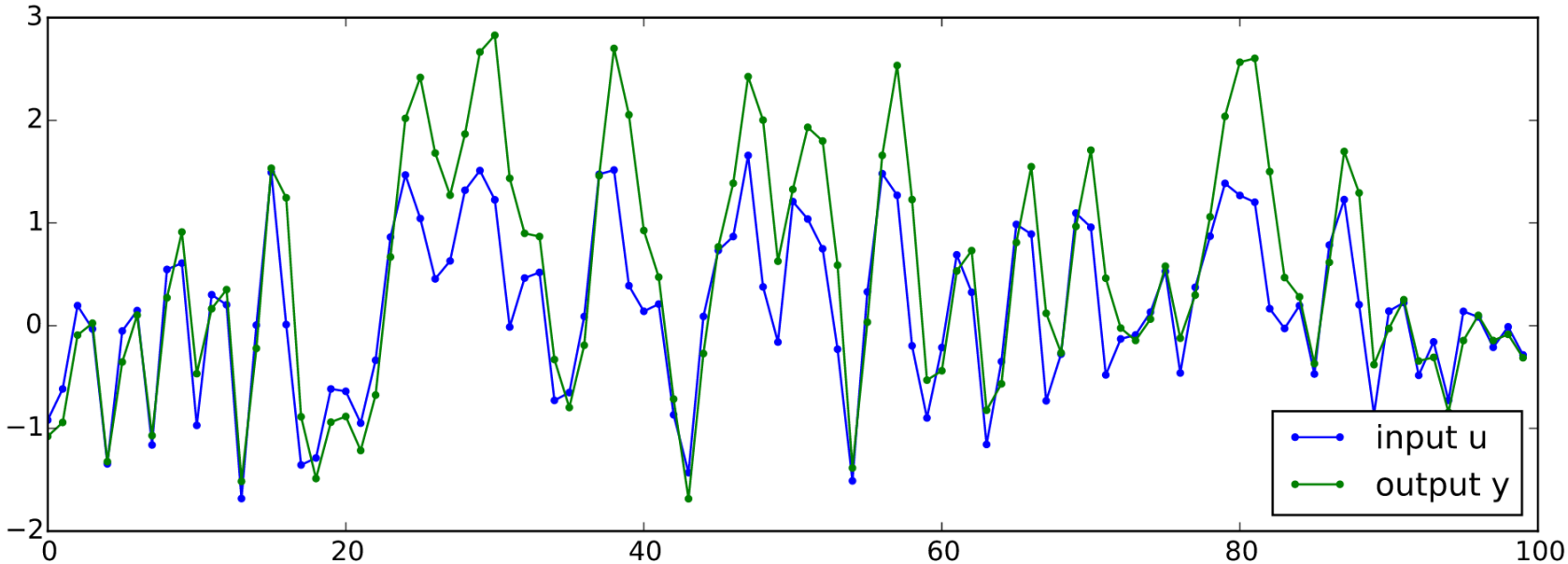
3. Using the backslash operator (similar to Matlab):

   `x = A\b`

   Note: Fastest and most reliable option!

Example: moving average model

- We are given a time series of input data $u_1, u_2, \ldots, u_T$ and output data $y_1, y_2, \ldots, y_T$. Example:



- A "moving average" model with window size k assumes each output is a weighted combination of k previous inputs:

$$y_t \approx w_1 u_t + w_2 u_{t-1} + \cdots + w_k u_{t-k+1} \qquad \text{for all } t$$

- find weights $w_1, \ldots, w_k$ that best agree with the data.

Example: moving average model

- Moving average model:

$$y_t \approx w_1 u_t + w_2 u_{t-1} + w_3 u_{t-2} \qquad \text{for all } t$$

- Writing all the equations (e.g. $k = 3$):

$$\begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ \vdots \\ y_T \end{bmatrix} \approx \begin{bmatrix} u_1 & 0 & 0 \\ u_2 & u_1 & 0 \\ u_3 & u_2 & u_1 \\ \vdots & \vdots & \vdots \\ u_T & u_{T-1} & u_{T-2} \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \\ w_3 \end{bmatrix}$$

- Solve least squares problem! Moving Average.ipynb