8. Least squares

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

Laurent Lessard (www.laurentlessard.com)
Review of linear equations

System of $m$ linear equations in $n$ unknowns:

\[
\begin{align*}
  a_{11}x_1 + \cdots + a_{1n}x_n &= b_1 \\
  a_{21}x_1 + \cdots + a_{2n}x_n &= b_2 \\
  \vdots & \quad \vdots \\
  a_{m1}x_1 + \cdots + a_{mn}x_n &= b_m
\end{align*}
\]

\[
\begin{bmatrix}
  a_{11} & \cdots & a_{1n} \\
  \vdots & \ddots & \vdots \\
  a_{m1} & \cdots & a_{mn}
\end{bmatrix}
\begin{bmatrix}
  x_1 \\
  \vdots \\
  x_n
\end{bmatrix}
= 
\begin{bmatrix}
  b_1 \\
  \vdots \\
  b_m
\end{bmatrix}
\]

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 $A$.

\[
Ax = \begin{bmatrix} a_1 & a_2 & \cdots & a_n \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = a_1x_1 + \cdots + a_nx_n = b
\]

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 have \( Ax \approx b \).

- The least-squares approach: make Euclidean norm \( \|Ax - b\| \) as small as possible.

- Equivalently: make \( \|Ax - b\|^2 \) as small as possible.

**Standard form:**

\[
\begin{align*}
\text{minimize} & \quad \|Ax - b\|^2 \\
\text{subject to} & \quad x_1, x_2, \ldots, x_n \in \mathbb{R}
\end{align*}
\]

It’s an unconstrained optimization problem.
Least squares

- Typical case of interest: $m > n$ (overdetermined). If there is no solution to $Ax = b$ we try instead to have $Ax \approx b$.

- The least-squares approach: make Euclidean norm $\|Ax - b\|$ as small as possible.

- Equivalently: make $\|Ax - b\|^2$ as small as possible.

Properties:

- $\|x\| = \sqrt{x_1^2 + \cdots + x_n^2} = \sqrt{x^Tx}$

- In Julia: $\|x\| = \text{norm}(x)$

- In JuMP: $\|x\|^2 = \text{dot}(x,x) = \text{sum}(x.^2)$
Least squares

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

\[
\|Ax - b\|^2 = \|(a_1x_1 + \cdots + a_nx_n) - b\|^2
\]
Least squares

- **row interpretation**: If $\tilde{a}_i^T$ is the $i^{th}$ row of $A$, define
  
  \[ r_i := \tilde{a}_i^T x - b_i \]
  
  to be the $i^{th}$ residual component.

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

We minimize the sum of squares of the residuals.

- Solving $Ax = b$ would make all residual components zero. Least squares attempts to make all of them small.
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.

Writing all the equations:

\[
\begin{align*}
y_1 & \approx px_1^2 + qx_1 + r \\
y_2 & \approx px_2^2 + qx_2 + r \\
\vdots \\
y_m & \approx px_m^2 + qx_m + r
\end{align*}
\]

\[
\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_m \end{bmatrix} \approx \begin{bmatrix} x_1^2 & x_1 & 1 \\ x_2^2 & x_2 & 1 \\ \vdots & \vdots & \vdots \\ x_m^2 & x_m & 1 \end{bmatrix} \begin{bmatrix} p \\ q \\ r \end{bmatrix}
\]

- Also called regression
Example: curve-fitting

- **More complicated**: \( y = p e^x + q \cos(x) - r \sqrt{x} + s x^3 \)
- 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 $r := b - Ax$. We can’t make it zero, so instead we try to make it *small*. Many options!

- **minimize the largest component (a.k.a. the $\infty$-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 \( \begin{bmatrix} x \\ x \end{bmatrix} \) that is closest to \( \begin{bmatrix} 1 \\ 2 \end{bmatrix} \).

Minimize largest component:

\[
\min_x \max\{|x - 1|, |x - 2|\}
\]

Optimum is at \( x = 1.5 \).
Example: find \[ \begin{bmatrix} x \\ x \end{bmatrix} \] that is closest to \[ \begin{bmatrix} 1 \\ 2 \end{bmatrix} \].

Minimize sum of components:

\[
\min_x |x - 1| + |x - 2|
\]

Optimum is any \( 1 \leq x \leq 2 \).
Vector norms

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

Minimize sum of squares:

\[
\min_x (x - 1)^2 + (x - 2)^2
\]

Optimum is at \[x = 1.5\].
Vector norms

- minimizing the largest component is an LP:
  \[
  \min_x \max_i |\tilde{a}_i^T x - r_i| \iff \min_{x,t} t \\
  \text{s.t. } -t \leq \tilde{a}_i^T x - r_i \leq t
  \]

- minimizing the sum of absolute values is an LP:
  \[
  \min_x \sum_{i=1}^m |\tilde{a}_i^T x - r_i| \iff \min_{x,t_i} t_1 + \cdots + t_m \\
  \text{s.t. } -t_i \leq \tilde{a}_i^T x - r_i \leq t_i
  \]

- minimizing the 2-norm is not an LP!
  \[
  \min_x \sum_{i=1}^m (\tilde{a}_i^T x - r_i)^2
  \]
The set of points \( \{Ax\} \) is a **subspace**.

We want to find \( \hat{x} \) such that \( A\hat{x} \) is closest to \( b \).

**Insight**: \( (b - A\hat{x}) \) must be orthogonal to all line segments contained in the subspace.
Geometry of LS

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

\[
A^TA\hat{x} = A^Tb
\]
Normal equations

**Theorem:** If \( \hat{x} \) satisfies the normal equations, then \( \hat{x} \) is a solution to the least-squares optimization problem

\[
\min_{x} \| Ax - b \|^2
\]

**Proof:** Suppose \( A^T A \hat{x} = A^T b \). Let \( x \) be any other point.

\[
\| Ax - b \|^2 = \| A(x - \hat{x}) + (A\hat{x} - b) \|^2
\]
\[
= \| A(x - \hat{x}) \|^2 + \| A\hat{x} - b \|^2 + 2(x - \hat{x})^T A^T (A\hat{x} - b)
\]
\[
= \| A(x - \hat{x}) \|^2 + \| A\hat{x} - b \|^2
\]
\[
\geq \| A\hat{x} - b \|^2
\]
Least squares problems are easy to solve!

- Solving a least squares problem amounts to 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:
   ```julia
   using JuMP
   m = Model()
   @variable( m, x[1:size(A,2)] )
   @objective( m, Min, sum((A*x-b).^2) )
   solve(m)
   **Note:** requires Mosek or Gurobi
   ```

2. Solving the normal equations directly:
   ```julia
   x = inv(A'*A)*(A'*b)
   **Note:** Requires A to have full column rank ($A^T A$ invertible)
   ```

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