Least Squares

Saturday, December 6, 2025 7:46 PM

Comment: Students have seen some aspects of least squares in

- APPM 3310 Matrix Methods
- Ch 8.1 in our text (Burden + Faires)

Thus these notes are part review, with some comments on <u>numerical implementation</u>

It's very common. Many names ("ordinary least squares", "regression", "linear regression")

many notations ($\vec{y} = X\vec{\beta} + \vec{z}$, $\vec{z} \sim N(0, \sigma^2)$ ... in statistics)

many fields (statistics, CS, all engineering, all social sciences, all natural sciences)

If you end up with a job as a data scientist, least squares is your 1^{st} method to try. If it works, great! If not, try a fancier model.

<u>Setup</u> Before

$$\begin{array}{cc} & n \\ n & \boxed{A} \end{array} \; \vec{x} = \vec{b}$$

Now, overdetermined

$$\begin{array}{cc} & n \\ m & \boxed{A} \end{array} \; \begin{array}{cc} n & \vec{x} \end{array} = \begin{array}{cc} m & \vec{b} \end{array} \qquad m > n$$

Not only can we not apply Gaussian elimination, but there's probably not even a solution.

<u>Instead</u>, we'll minimize the residual

Least Squares: Find $\vec{x}$ that minimizes $\| A\vec{x} - \vec{b} \|_2$ ← Euclidean norm

i.e., find $\vec{x} = \underset{\vec{x}}{\operatorname{argmin}} \| A\vec{x} - \vec{b} \|_2$

Note: equivalent to $\vec{x} = \operatorname{argmin} \frac{1}{2} \| A\vec{x} - \vec{b} \|_2^2$

or $\vec{x} = \operatorname{argmin} \frac{1}{m} \| A\vec{x} - \vec{b} \|_2^2$

(constants, monotonic transformations don't affect the solution)

p. 2

Saturday, December 6, 2025 7:46 PM

Computing the solution, method 1 (not recommended for ill-conditioned cases): the normal equations

want $\vec{x} \in \underset{\vec{x}}{\operatorname{argmin}} \left(F(\vec{x}) := \frac{1}{2}\|A\vec{x}-\vec{b}\|^2 \right)$

Fact: $\nabla F(\vec{x}) = A^T(A\vec{x}-\vec{b})$ (a vector)

Fact: F is convex and differentiable

Fact: If F is convex and differentiable, then the solution(s) to the unconstrained minimization problem $\min_{x} F(\vec{x})$ can be found by solving $\underbrace{\nabla F(\vec{x}) = 0}_{\text{i.e. necessary and sufficient}}$

So, to find $\vec{x} = \underset{\vec{x}}{\operatorname{argmin}} \frac{1}{2}\|A\vec{x}-\vec{b}\|^2$, we solve

$$A^T(A\vec{x}-\vec{b}) = 0 \quad \text{i.e.} \quad \boxed{A^TA\,\vec{x} = A^T\vec{b}}$$ the "normal equations"

This is a square system, often invertible, so we can solve it!

--- but, $\kappa_2(A^TA) = \kappa_2(A)^2$ so we'll lose more digits of accuracy than we needed to

Computing the solution, method 2 (better)

Do a QR-decomposition of A, $\underset{m \times n}{A} = \underset{m \times n}{Q}\;\underset{n \times n}{R}$

Q: (partial) orthogonal, $Q^TQ = I_{n\times n}$

R: upper triangular

you can do this via, e.g., Gram-Schmidt
or its stable variant modified Gram-Schmidt
(or, better, just as Matlab or Scipy for it) or better yet, Householder

then if A is full rank; $\underbrace{\mathrm{col}}_{\text{column span}}(A) = \mathrm{col}(Q)$ and R is non-singular

p. 3

Saturday, December 6, 2025 7:46 PM

thin QR: $Q^TQ = I$

[R^{-1} exists if A is full column rank]

Fact (partial derivation in demo): if $A = QR$

$$\underset{\vec{x}}{\operatorname{argmin}} \|A\vec{x} - \vec{b}\| = \underset{\vec{x}}{\operatorname{argmin}} \|R\vec{x} - Q^T\vec{b}\| = \{\vec{x} : R\vec{x} = Q^T\vec{b}\}$$

so just solve the $n \times n$ upper triangular system $R\vec{x} = Q^T\vec{b}$ using back-substitution.

One way to think of it: the normal equations $A^TA\vec{x} = A^T\vec{b}$ plug in $A = QR$

$$R^T\cancel{Q^T}\cancel{Q}R\vec{x} = R^TQ^T\vec{b}$$

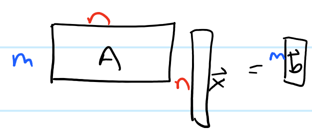
$$R^{-T}(R^TR\vec{x} = R^TQ^T\vec{b})$$

so... $R\vec{x} = Q^T\vec{b}$

} doing $Q^TQ = I$ cancellation "by pen and paper" helps make it more stable...

Method 3: SVD won't go into details, but also reasonable

underdetermined systems can also be solved by an approach you might call "least-squares" also



$m < n$

If a solution exists (it will if A is full rank, i.e., rank(A) = m)

then an infinite number of solutions exist! So which one to choose?

Often prefer this one:

$$\vec{x} = \operatorname{argmin} \|\vec{x}\|_2 \quad \text{s.t. } Ax = b$$

Computationally, $\vec{x} = A^{+}\vec{b}$ ← means pseudo-inverse in numerics

$:= A^T(AA^T)^{-1}$

pinv(A) in Matlab

In Matlab, if A is square, A\b solves $A\vec{x} = \vec{b}$

over-determined A\b solves $\min \|A\vec{x} - \vec{b}\|^2$

under-determined A\b finds a solution to $A\vec{x} = \vec{b}$ that has many zeros.

... so not $\underbrace{\min \|\vec{x}\|_2 \text{ st } A\vec{x} = \vec{b}.}_{\text{for this, do pinv(A) * b}}$

In Python, don't confuse these two:

- scipy.linalg.lstsq <-- linear least squares (what we've been talking about today). Pure linear algebra.
- scipy.optimize.least_squares <-- nonlinear least squares, doesn't guarantee global solution. Optimization