Least Squares

Thursday, December 3, 2020 11:32 PM

This will be covered somewhat in ch 8.1 (Burden + Faires), 2^{nd} semester

If you're not taking 2^{nd} semester, you need to know about least-squares

It's very common. Many names ("ordinary least squares", "regression", "linear regression")
many notations ($y = X\beta + z$, $z \sim N(0,\sigma^2)$ ... 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.

Setup

Before

$$n\overset{n}{\boxed{\;A\;}}\;\underset{x}{\boxed{\,}} = \underset{b}{\boxed{\,}}$$

Now, overdetermined

$$m\overset{n}{\boxed{\;A\;}}\;n\boxed{x} = m\boxed{\,}\;b \qquad m > n$$

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

Instead, 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)

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

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

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

Fact: F is convex and differentiable

Fact: If F is convex and differentiable, then the solution(s) to the unconstrained minimization problem $\min_x F(x)$ can be found by solving $\underbrace{\nabla F(x) = 0}$

i.e. necessary and sufficient

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

$$A^T(Ax-b) = 0 \quad \text{i.e.} \quad \boxed{A^TA\,x = A^Tb}$$ 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, $\;{}_m\!\left[\,A\,\right]^{n} = {}_m\!\left[\,Q\,\right]^{n}\;{}_n\!\left[\,R\,\right]^{n}$

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

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)

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

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

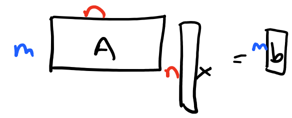
$$\operatorname{argmin} \|Ax-b\| = \operatorname{argmin} \|Rx - Q^Tb\| = \{x : Rx = Q^Tb\}$$

so just solve the $n \times n$ upper triangular system $Rx = Q^Tb$ using back-substitution.

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., $\text{rank}(A) = m$) then an infinite number of solutions exist! So which one to choose?

Often prefer this one:

$$x = \arg\min \|x\|_2 \quad \text{s.t. } Ax = b$$

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

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

pinv(A) in Matlab

⚠ In Matlab, if A is square, A\b solves $Ax=b$

over-determined A\b solves $\min \|Ax-b\|^2$

under-determined A\b finds a solution to $Ax=b$ that has many zeros.

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