Linear Algebra and Least Squares Refresher

Arnold Kalmbach, Yewon Lee CSC477 Tutorial #2 Sept 28, 2022

Outline

- Notation reminder, and a couple useful facts
- Intro to linear least squares (with an example)
- Singular Value Decomposition
 - Relationship to eigendecomposition

Notation Reminder - Inverse

$$AX = I \iff X = A^{-1}$$

- If A⁻¹ exists, A is nonsingular.
- All the following are equivalent
 - \circ **A** is nonsingular
 - o det(A) != 0
 - o rank(A) = n
 - Ax = 0 has a unique solution x = 0

Notation Reminder - Orthogonal Matrix

$$Q^T Q = Q Q^T = I$$

• Equivalently, if all the rows & columns are orthonormal ie

$$Q = \begin{bmatrix} q_1, q_2, \dots, q_n \end{bmatrix}$$
$$q_i^T q_j = \begin{cases} 1 & i = j \\ 0 & i \neq j \end{cases}$$

Notation Reminder - Vector Norms $\|v\|$

• P-Norms

$$\|v\|_{p} = \left(\sum_{i=1}^{n} |v_{i}|^{p}\right)^{1/p}$$

 1-Norm = Sum of elements, 2-Norm = Euclidean Distance, Infinity-Norm = Largest Element

Linear Least Squares by Example

How to Estimate the Location of the Wall?



Idea: Fit a line to the sample points.

This is an over-determined system

Least Squares: Minimize Sum of Squared Error



Least Squares: Minimize Sum of Squared Error



Least Squares: Minimize Sum of Squared Error



Least Squares Criterion

Very general formulation: $\hat{x} = \arg \min_{x} \|b - h_x(A)\|_2^2$

Most common options (not covered much in this course):

- 1. Take gradient of f(x) wrt. x
- 2. Approximate *h* with a linear function h = A x

Can use Linear or nonlinear least squares to set up all kinds of modelling problems as optimization problems.

Least Squares: Solution

$$\nabla f(\hat{x}) = 2A^T (A\hat{x} - b) = 0$$
$$(A^T A)\hat{x} = A^T b$$
$$\hat{x} = (A^T A)^{-1} A^T b = A^{\dagger} b$$

If $b = Ax^{\star} + v$, $v \sim \mathcal{N}(0, \sigma I)$ this produces the **optimal** estimator (BLUE)

To calculate with numpy:

- numpy.linalg.pinv(A)
- x_hat = numpy.linalg.lstsq(A, b)

Issues Computing the Solution

$$\hat{x} = \left(A^T A\right)^{-1} A^T b = A^{\dagger} b$$

Computing $(A^T A)^{-1}$ by Gaussian Elimination is numerically unstable and slow!

We can do better if we decompose A

• $A = LL^T$ (Cholesky Factorization, **skip**) • $A = \begin{bmatrix} Q_1 & Q_2 \end{bmatrix} \begin{bmatrix} R \\ 0 \end{bmatrix}$ (QR Factorization, **skip**) • $A = U\Sigma V^T$ (Singular Value Decomposition)

Singular Value Decomposition (SVD)



Singular Value Decomposition (SVD)



 \sum^{\dagger} = Reciprocal of each diagonal entry, transpose

Very useful fact:

$$A^T A = V \Sigma^{\dagger} U^T U \Sigma V^T = V (\Sigma^{\dagger} \Sigma) V^T$$

SVD Example



1

 Σ has diagonal elements

$$\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_k > \sigma_{k+1} = \sigma_{k+2} = \ldots = 0$$

Where σ_i are the square root of the eigenvalues of $A^T A$ and $k = \operatorname{rank}(A)$

$$egin{aligned} A &= egin{bmatrix} 1 & 0 \ \end{bmatrix} \ A^T A &= egin{bmatrix} 1 & 0 \ \end{bmatrix} &= egin{bmatrix} 1 & 0 \ \end{bmatrix} = egin{bmatrix} 1 & 0 \ \end{bmatrix} \ \det(A^T A - \lambda I) &= 0 \Rightarrow \lambda_1 = 1, \ ext{rank}(A) = \ \Sigma &= egin{bmatrix} 1 & 0 \ \end{bmatrix} \end{aligned}$$

SVD Examplenxdnxn nxddxd
$$V = [v_1 \ v_2 \ \dots \ v_k \ v_{k+1} \ \dots \ v_d]$$
 $V = [v_1 \ v_2 \ \dots \ v_k \ v_{k+1} \ \dots \ v_d]$ Normalized
eigenvectors of
 $A^T A$ Obtained from $A^T A v_j = 0$
such that orthogonality of V is
satisfied

$$egin{aligned} & (A^TA-\lambda_1I)v_1=0,\ \lambda_1=1 & & A^TAv_2=0 \ & & \left[egin{aligned} 1 & 0 \ 0 & -1 \end{array}
ight] v_1=0 & & & \left[egin{aligned} 1 & 0 \ 0 & 0 \end{array}
ight] v_2=0 & & & \Rightarrow V=\left[egin{aligned} 1 & 0 \ 0 & 1 \end{array}
ight] & & v_1\cdot v_2=0 & & & \Rightarrow V=\left[egin{aligned} 1 & 0 \ 0 & 1 \end{array}
ight] & & v_2=\left[egin{aligned} 1 & 0 \ 0 & 1 \end{array}
ight] & & v_2=\left[egin{aligned} 1 & 0 \ 0 & 1 \end{array}
ight] & & v_2=\left[egin{aligned} 1 & 0 \ 0 & 1 \end{array}
ight] & & v_2=\left[egin{aligned} 1 & 0 \ 0 & 1 \end{array}
ight] & & v_2=\left[egin{aligned} 1 & 0 \ 0 & 1 \end{array}
ight] & & v_2=\left[egin{aligned} 1 & 0 \ 1 & 0 \end{array}
ight] & & v_2=\left[egin{aligned} 1 & 0 \ 0 & 1 \end{array}
ight] & & v_2=\left[egin{aligned} 1 & 0 \ 0 & 1 \end{array}
ight] & & v_2=\left[egin{aligned} 1 & 0 \ 1 & 0 \end{array}
ight] & & v_2=\left[egin{aligned} 1 & 0 \ 1 & 0 \end{array}
ight] & & v_2=\left[egin{aligned} 1 & 0 \ 1 & 0 \end{array}
ight] & & v_2=\left[egin{aligned} 1 & 0 \ 1 & 0 \end{array}
ight] & & v_2=\left[egin{aligned} 1 & 0 \ 1 & 0 \end{array}
ight] & & v_2=\left[egin{aligned} 1 & 0 \ 1 & 0 \end{array}
ight] & & v_2=\left[egin{aligned} 1 & 0 \ 1 & 0 \end{array}
ight] & & v_1=\left[egin{aligned} 1 & 0 \ 0 & 0 \end{array}
ight] & & v_2=\left[egin{aligned} 1 & 0 \ 1 & 0 \end{array}
ight] & & v_2=\left[egin{aligned} 1 & 0 \ 1 & 0 \end{array}
ight] & & v_1=\left[egin{aligned} 1 & 0 \ 0 & 0 \end{array}
ight] & & v_2=\left[egin{aligned} 1 & 0 \ 1 & 0 \end{array}
ight] & & v_2=\left[egin{aligned} 1 & 0 \ 0 & 0 \end{array}
ight] & & v_2=\left[egin{aligned} 1 & 0 \ 0 & 0 \end{array}
ight] & & v_2=\left[egin{aligned} 1 & 0 \ 0 & 0 \end{array}
ight] & & v_2=\left[egin{aligned} 1 & 0 \ 0 & 0 \end{array}
ight] & & v_2=\left[egin{aligned} 1 & 0 \ 0 & 0 \end{array}
ight] & & v_2=\left[egin{aligned} 1 & 0 \ 0 & 0 \end{array}
ight] & & v_2=\left[egin{aligned} 1 & 0 \ 0 & 0 \end{array}
ight] & & v_2=\left[egin{aligned} 1 & 0 \ 0 & 0 \end{array}
ight] & & v_1=\left[egin{aligned} 1 & 0 \ 0 & 0 \end{array}
ight] & & v_2=\left[egin{aligned} 1 & 0 \ 0 & 0 \end{array}
ight] & & v_1=\left[egin{aligned} 1 & 0 \ 0 & 0 \end{array}
ight] & & v_2=\left[egin{aligned} 1 & 0 \ 0 & 0 \end{array}
ight] & & v_2=\left[egin{aligned} 1 & 0 \ 0 & 0 \end{array}
ight] & & v_2=\left[egin{aligned} 1 & 0 \ 0 & 0 \end{array}
ight] & & v_1=\left[egin{aligned} 1 & 0 \ 0 & 0 \end{array}
ight] & & v_2=\left[egin{aligned} 1 & 0 \ 0 & 0 \end{array}
ight]$$

SVD Example
$$A = U \Sigma V^T$$

 $U = \begin{bmatrix} u_1 \ u_2 \ \dots \ u_k \ u_{k+1} \ \dots \ u_n \end{bmatrix}$
Normalized AA^T Obtained from $AA^T u_j = 0$

$$egin{aligned} AA^T &= egin{bmatrix} 1 & 0 \end{bmatrix} egin{bmatrix} 1 \ 0 \end{bmatrix} &= 1 \ \det(AA^T - \lambda I) &= 0 \Rightarrow \lambda_1 = 1 \ AA^T u_1 &= \lambda_1 u_1 \iff u_1 = u_1 \ u_1 &= 1 \ U &= 1 \end{aligned}$$

If we can compute $A^{\dagger} = (A^T A)^{-1} A^T$ stably, we can solve LS problems.

Recall:
$$A^T A = V \left(\Sigma^\dagger \Sigma \right) V^T$$
 So $A^\dagger = V \Sigma^\dagger U^T$

To calculate with numpy

• U,S,V = numpy.linalg.svd(A)