Lecture 4: Linear Least Squares

CSC 338: Numerical Methods

Ray Wu

University of Toronto

February 1, 2023

Overview

- Linear Least Squares
- Normal Equations and Derivation
- Application: Data fitting
- QR Decomposition
- Singular Value Decomposition
- Image compression

Linear Least Squares

- Last lecture, we focused on

$$Ax = b \tag{1}$$

  when A is a square matrix.
- This lecture: what if A is not a square matrix? Example:

$$\begin{bmatrix} 3 & 4 \\ 1 & 7 \\ 2 & 5 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 4 \\ 6 \\ 3 \end{bmatrix} \tag{2}$$

- Least squares compute an approximate solution to these linear systems, by minimizing the residual $r = b - Ax$ in the 2-norm.

$$\min_x \|b - Ax\| \tag{3}$$

Why the 2-norm?

- The choice of norm relates to how we "count" the distance.
- Alternative norms:
  - 1-norm: $\min \|b - Ax\|_1$. Used in least absolute deviations.
  - max-norm: $\min \|b - Ax\|_\infty = \min \max_i |b - Ax|$
- Both 1-norm and max-norm problems lead to linear programming (linear optimization) problems.
  - Simplex algorithm, IPMs (e.g. Karmarkar 1984), etc.
  - Beyond the scope of this course. You may find coverage in optimization, machine learning, or theoretical computer science.
- 2-norm leads to simple solutions
- maximum likelihood estimate (MLE):
  - 2-norm leads to the MLE for normal distributions, which are ubiquitous in modelling
  - 1-norm leads to the MLE for double exponential (Laplace) distributions
- 1-norm is robust to outliers

Normal Equations – Derivation

Two derivations:

- Define $\phi(x) = \|b - Ax\|^2$, set derivatives to zero.
- Using geometry and orthogonality.

Normal Equations

- Two vectors u and v are **orthogonal** if and only if $u^T v = 0$.
- Recall that we wish to minimize $\|b - Ax\|$.
- Find $y = Ax$ that is the closest vector in $\text{col}(A)$ to b.
- Want residual to be orthogonal to every vector in a spanning set of that space.
- Therefore, $r = b - Ax$ is orthogonal to every column of A.

$$\forall i, a_i^T(b - Ax) = 0 \tag{4}$$

or in other words (matrix notation),

$$A^T(b - Ax) = \vec{0} \tag{5}$$

- Rearranging, we get the normal equations:

$$A^T Ax = A^T b \tag{6}$$

Normal Equations visualized

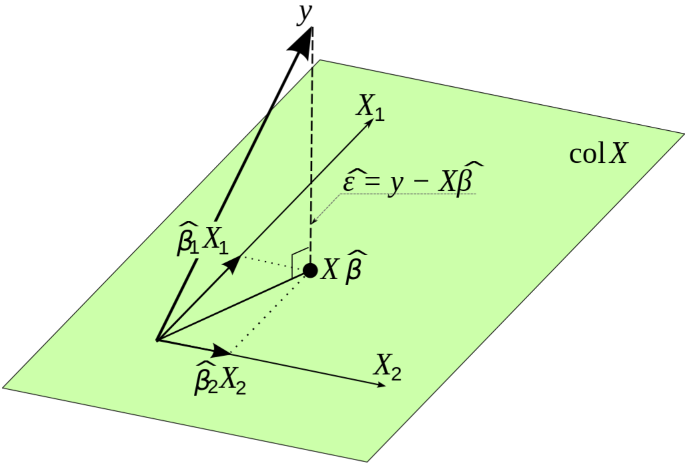


Figure 1: Visualization of Geometric Interpretation of Least Squares. Source: Wikimedia Commons.

Normal Equations (II)

- From last slides,

$$A^T(b - Ax) = 0 \tag{7}$$

- Normal equation method:
  1. Compute $A^T A$ and $A^T b$.
  2. Decompose $A^T A$ using Cholesky factorization and use forward/backward solves for triangular systems.

Application: data fitting

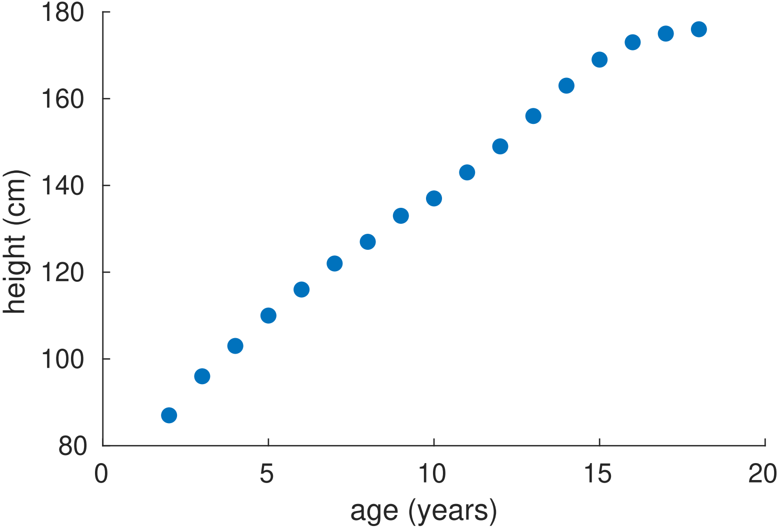
- Suppose we have observed data $(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)$
- We want to fit this to some model, for example, $y = ax + b$.
- Create n equations with each pair of (x_i, y_i).
- Solve the resulting overdetermined system of linear equations.

Data fitting example

Age	Height
2	87
3	96
4	103
⋮	⋮
18	176

Table 1: Median height of male children in Canada

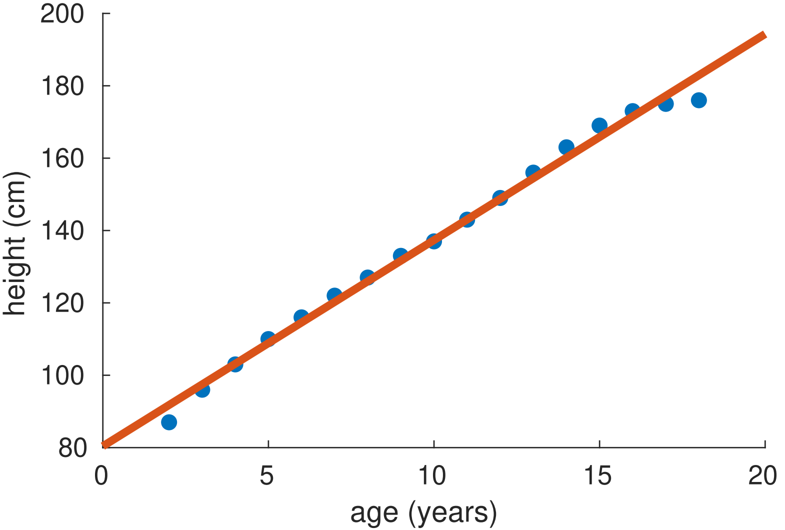
Looking at the data



Setting up the equations

$$\begin{bmatrix} 1 & x_1 \\ 1 & x_2 \\ \vdots & \vdots \\ 1 & x_n \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} \tag{8}$$

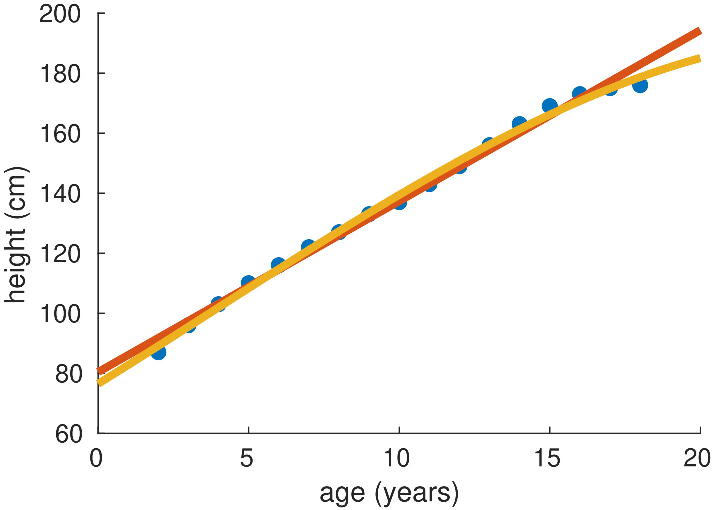
Linear regression model



With nonlinear functions

$$\begin{bmatrix} 1 & x_1 & x_1^2 & x_1^3 \\ 1 & x_2 & x_2^2 & x_2^3 \\ \vdots & \vdots & \vdots & \vdots \\ 1 & x_n & x_n^2 & x_n^3 \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \\ c_3 \\ c_4 \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} \tag{9}$$

Linear regression with nonlinear functions



Issues with Normal equation method

Suppose we have the matrix

$$A = \begin{bmatrix} 1 & 1 & 1 \\ \epsilon & 0 & 0 \\ 0 & \epsilon & 0 \\ 0 & 0 & \epsilon \end{bmatrix} \tag{10}$$

for some small value of ϵ, say 10^{-10}. Then, symbolically,

$$A^T A = \begin{bmatrix} 1+\epsilon^2 & 1 & 1 \\ 1 & 1+\epsilon^2 & 1 \\ 1 & 1 & 1+\epsilon^2 \end{bmatrix} \tag{11}$$

- Numerically, $A^T A$ is singular (and the calculations cannot continue), but A has full rank.
- Is this an issue of the problem, or an issue of the algorithm?

Singular Value Decomposition

- The **singular value decomposition** decomposes a general matrix A into the form
$$A = U\Sigma V^T \tag{12}$$
where U and V are orthogonal matrices, and Σ is a diagonal matrix, with the diagonal entries called the *singular values*.
  - The SVD always exists, and is not unique. By convention, we arrange Σ such that the singular values are sorted and the largest singular value is in the $(1, 1)$ location.
  - The SVD is a generalization of the eigendecomposition of a matrix (i.e. $A = MDM^{-1}$).
- Since multiplication with orthogonal matrices do not change norm, we have
$$\|A\| = \|\Sigma\| = \sigma_1 \tag{13}$$
- Now consider the pesudoinverse of A: the norm is is $\frac{1}{\sigma_n}$, hence the condition number of A is $\frac{\sigma_1}{\sigma_n}$.

Condition number of normal equations

- The condition number of an symmetric positive definite matrix $B = A^T A$ is given by the ratio between its largest and smallest eigenvalues. This is equivalent to

$$\kappa(B) = \frac{\lambda_1}{\lambda_n} = \frac{\sigma_1^2}{\sigma_n^2} = \kappa(A)^2. \tag{14}$$

- because

$$B = A^T A = (U\Sigma V^T)^T U\Sigma V^T = V\Sigma U^T U\Sigma V^T = V\Sigma^2 V^T \tag{15}$$

giving a diagonalization of B.

- Hence, constructing the normal equations squares the condition number. So this is an issue of the algorithm, and not the problem.
- This means we should look for alternatives to the normal equation method.

QR decomposition and Householder reflections

- Suppose we have a decomposition

$$A = Q \begin{bmatrix} R \\ 0 \end{bmatrix} \tag{16}$$

for some orthogonal matrix Q and a triangular matrix R. Then we have

$$\|b - Ax\| = \|b - Q \begin{bmatrix} R \\ 0 \end{bmatrix} x\| = \|Q^T b - \begin{bmatrix} R \\ 0 \end{bmatrix} x\|. \tag{17}$$

- To minimize this expression, we rewrite $Q^T b$ as $[c \quad d]^T$, where c has the same number of entries as Rx and d has the remaining entries.

$$\|Q^T b - \begin{bmatrix} R \\ 0 \end{bmatrix} x\| = \| \begin{bmatrix} c - Rx \\ d \end{bmatrix} \| \tag{18}$$

- We have no control over d, so we solve the system $Rx = c$ to minimize the other components.

Householder reflections

- We want to decompose $A = QR$, so, we need to find orthogonal transformations that transform A into an upper triangular matrix R.
- The idea is to apply a sequence of orthogonal transformations that zero out the matrix entries that we want to.
- Consider the matrix $P = I - 2uu^T$, for an arbitrary unit vector u. Now, we want to find the right u such that $Pz = \alpha e_1$.
- What do we know about P? P is a reflection across the plane defined by the normal vector u.
  - $Pu = u - 2u(u^T u) = -u$
  - $Pv = v - 2u(u^T v) = v$ if v is orthogonal to u.

Householder reflections (II)

- Write $Pz = z - 2uu^T z = z - (2u^T z)u = \alpha e_1$
- Then u is the unit vector in the direction $z - \alpha e_1$ (rearrange and divied by $2u^T z$).
- Since P is an orthogonal transformation (assignment question), $\|Pz\| = \|z\|$ and hence $\alpha = \|z\|$.
- Therefore, $u = z \pm \|z\| e_1$ (In practice, pick the same sign as the first entry of z, to avoid any possibilty of cancellation error.)
- Finally, we apply householder reflections to zero out all entries below the i, i-th entry of A, and complete our orthogonal transformation.
- The series of reflections is the matrix Q, and the resulting matrix is R.

Recall that the singular value decomposition is given by

$$A = U\Sigma V^T \tag{19}$$

Hence,

$$\|b - Ax\| = \|U^T b - \Sigma V^T x\| \tag{20}$$

- If the condition number is not too large, then we can directly solve the system.

$$U^T b - \Sigma V^T x \tag{21}$$

- Kind of defeats the purpose of SVD, since QR will also work.
- QR is faster to compute than SVD (We will not get into computing SVD).
- The real benefit of SVD occurs when A is not numerically full rank.

Singular Value Decomposition, Part 2

- If A is not full rank numerically, then the ratio σ_1/σ_n is very large ($> 10^{16}$).
- Solution: remove the singular values that are too small.
- Starting from n and going backwards, find a value r such that σ_1/σ_r is acceptable, and set the remaining singular values to zero.
- Truncate the matrices U and V to only store the first r rows/columns.
- A is compressed from $m \times n$ into $r(m + n + 1)$ storage locations
- This is a rank-r approximation of the matrix A. In fact, it is the *best* rank-r approximation, as measured by the Frobinus norm.

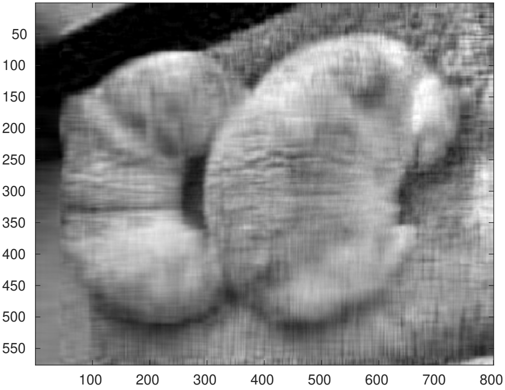
Example - Image compression

Consider this cat with the croissant, with the pixels stored as real numbers in a matrix A:

There are $800 \times 576 = 460800$ entries we have to store in grayscale.

Example - Image compression

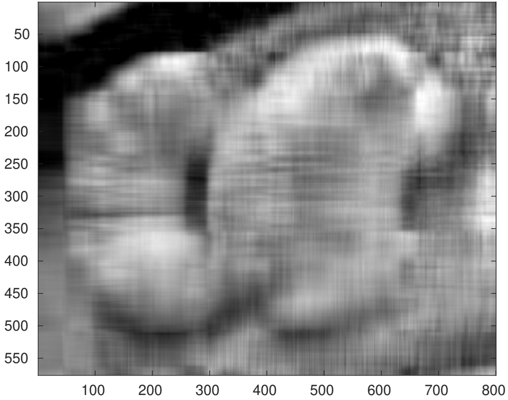
Rank-20 approximation of A:



We only need to store $20 \times (800 + 576 + 1) = 27540$ entries.

Example - Image compression

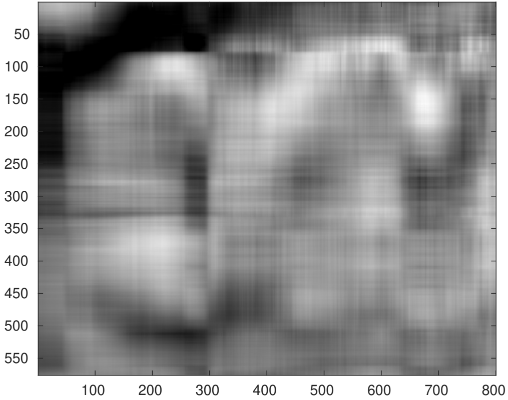
Rank-10 approximation of A:



We only need to store $10 \times (800 + 576 + 1) = 13770$ entries.

Example - Image compression

Rank-5 approximation of A:



We only need to store $5 \times (800 + 576 + 1) = 6885$ entries.