Notes

- Simpler right-looking derivation (sorry):

$$\begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} = \begin{bmatrix} 1 & \\ L_{21} & L_{22} \end{bmatrix} \begin{bmatrix} U_{11} & U_{12} \\ & U_{22} \end{bmatrix}$$
$$= \begin{bmatrix} U_{11} & U_{12} \\ L_{21}U_{11} & L_{21}U_{12} + L_{22}U_{22} \end{bmatrix}$$

$$U_{11} = A_{11}$$
$$U_{12} = A_{12}$$
$$L_{21} = A_{21} / A_{11}$$
$$L_{22}U_{22} = A_{22} - L_{21}U_{12}$$

From Last Time

- Solving linear least squares:
$$\min_x \|b - Ax\|_2^2$$
- Normal equations:
$$A^T A x = A^T b$$
  - Potentially unreliable if A is “ill-conditioned” (columns of A are close to being linearly dependent)
- Can we solve the problem more reliably?

The Best A

- Start by asking what is the best A possible?
- $A^T A = I$ (the identity matrix)
  - I.e. the columns of A are orthonormal
- Then the solution is $x = A^T b$, no system to solve (and relative error behaves well)
- What if A is not orthonormal?
- Change basis to make it so…

Orthonormalizing A

- Goal: find R so that A=QR
  - Q is orthonormal
  - R is easy to solve with
$$\|b - Ax\|_2^2 = \|b - QRx\|_2^2$$
$$= \|b - Qy\|_2^2, \quad Rx = y = Q^T b$$
- Classic answer: apply Gram-Schmidt to columns of A (R encodes the sequence of elementary matrix operations used in GS)

Gram-Schmidt

- Classic formula: $q_i = A_{*i} - \sum_{j=1}^{i-1} Q_{*j} \left(Q_{*j}^T A_{*i} \right)$
$$Q_{*i} = \frac{1}{\sqrt{q_i^T q_i}} q_i$$
- In-depth numerical analysis shows error (loss of orthogonality) can be bad
- Use Modified Gram-Schmidt instead:
  $q_i = A_{*i}$
  for j=1:i-1
  $\quad q_i = q_i - Q_{*j}(Q_{*j}^T q_i)$

What is R?

- Since A=QR, we find $R = Q^T A$
- Upper triangular, and containing exactly the dot-products from Gram-Schmidt
- Triangular matrices are easy to solve with: good!
- In fact, this gives an alternative to solving regular linear systems: A=QR instead of A=LU
  - Potentially more accurate, but typically slower

Another look at R

- Since A=QR, we have $A^TA=R^TQ^TQR=R^TR$
- That is, R^T is the Cholesky factor of A^TA
- But Cholesky factorization is not a good way to compute it!

Yet another look at R

- There is an even better way to compute R (than Modified Gram-Schmidt): orthogonal transformations
- Idea: instead of upper-triangular elementary matrices turning A into Q, use orthogonal elementary matrices to turn A into R
- Two main choices:
  - Givens rotations: rotate in selected two dimensions
  - Householder reflections: reflect across a plane

Givens rotations

- For $c^2+s^2=1$:

$$Q = \begin{pmatrix} I & 0 & 0 & 0 & 0 \\ 0 & \boxed{c} & 0 & \boxed{s} & 0 \\ 0 & 0 & I & 0 & 0 \\ 0 & \boxed{-s} & 0 & \boxed{c} & 0 \\ 0 & 0 & 0 & 0 & I \end{pmatrix}$$

- Say we want QA to be zero at (i,j):

$$sA_{jj} = cA_{ij}$$

Householder reflections

- For a unit vector v (normal to plane):

$$Q = I - 2vv^T$$

- Choose v to zero out entries below the diagonal in a column
- Note: can store Householder vectors and R in-place of A
  - Don't directly form Q, just multiply by Householder factors when computing Q^Tb

Full and Economy QR

- Even if A is rectangular, Givens and Householder implicitly give big square Q (and rectangular R): called the "full QR"
  - But you don't have to form the big Q
  - If you do: the extra columns are an orthonormal basis of the null-space of A^T
- Modified Gram-Schmidt computes only the first k columns of Q (rectangular Q) and gives only a square R: called the "economy QR"

Weighted Least Squares

- What if we introduce nonnegative weights (some data points count more than others)

$$\min_x \sum_{i=1}^n w_i \left(b_i - (Ax)_i\right)^2$$

$$\min_x (b - Ax)^T W (b - Ax)$$

- Weighted normal equations:

$$A^T W A x = A^T W b$$

- Can also solve with

$$\sqrt{W} A = QR$$

Moving Least Squares (MLS)

- Idea: estimate f(x) by fitting a low degree polynomial to data points, but weight nearby points more than others
- Use a weighting kernel W(r)
  - Should be big at r=0, decay to zero further away
- At each point x, we have a (small) weighted linear least squares problem:

$$\min_p \sum_{i=1}^{n} W\left(\|x - x_i\|\right)\left[f_i - p(x_i)\right]^2$$

Constant Fit MLS

- Instructive to work out case of zero degree polynomials (constants)
- Sometimes called Franke interpolation
- Illustrates effect of weighting function
  - How do we force it to interpolate?
  - What if we want local calculation?

MLS Basis

- Going back to general least squares problem:

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

- Solution x depends linearly on data b
- Add or scale data, solutions add and scale
- Same is true for MLS
- Can solve MLS for (0,…,0,1,0,…,0) etc. to get a set of basis functions