Week 6: Wednesday, Sep 26

Logistics

1. HW 2 is graded.
2. HW 3 is posted. Because October 8 is Fall Break, it is due on Wednesday, October 10.
3. I have an A-exam at 9 am next Friday, October 5. Consequently, I will have to skip office hours on that day.

Linear least squares

Suppose $A \in \mathbb{R}^{m \times n}$ where $m > n$. Then in general we will not be able to solve systems of the form $Ax = b$, and the best we can do is to minimize the residual error. Minimizing in the 2-norm gives us the standard least squares problem:

$$\operatorname{argmin}_x \|Ax - b\|_2^2.$$

Thing of the squared residual as a quadratic function in x:

$$F(x) = \|Ax - b\|^2 = (Ax - b)^T (Ax - b) = x^T A^T A x - 2x^T A^T b + b^T b.$$

Then the minimum occurs when

$$\nabla F(x) = 2(A^T A x - A^T b) = 0.$$

Thus we have

$$A^T A x = A^T b.$$

These are the *normal equations*, so named because they are exactly the equations that make the residual $Ax - b$ orthogonal (normal) to anything vector Ay in the range space of A.

If A is full rank, then $A^T A$ is symmetric and positive definite matrix, and the normal equations have a unique solution that we can compute via Cholesky factorization. But $\kappa(A^T A) = \kappa(A)$, so if $\kappa(A)$ is even moderately large, the condition number for the normal equations may be terrible. We will therefore t

If A is rank-deficient, we have a *rank-deficient least squares* problem. Though (nearly) rank-deficient least squares problems are fairly common in practice, for the moment we will concentrate on the case when A has full rank.

Orthogonal transformations and Gram-Schmidt

Recall that orthogonal transformations have the property (and indeed can be defined by the property) that they leave the Euclidean norm alone. If Q is any matrix with orthonormal columns, we can write

$$\|Ax - b\|_2^2 = \|Q^T(Ax - b)\|_2^2 = \|Q^T Ax - Q^T b\|_2^2.$$

This suggests an alternative approach to the least squares problem: find Q such that $Q^T A$ has a relatively simple form. A natural choice is the decomposition

$$A = QR,$$

where Q is an $m \times m$ orthogonal matrix and R is an $m \times n$ upper triangular matrix. Eqvuivalently, we can write the "economy" version of the decomposition, $A = QR$ with an $m \times n$ matrix Q and an $n \times n$ upper triangular R, where the columns of Q form an orthonormal basis for the range space of A. Using this decomposition, we can solve the least squares problem via the triangular system

$$Rx = Q^T b.$$

The *Gram-Schmidt* procedure is usually the first method people learn for converting some existing basis (columns of A) into an orthonormal basis (columns of Q). For each column of A, the procedure subtracts off any components in the direction of the previous columns, and then scales the remainder to be unit length. In MATLAB, Gram-Schmidt looks something like this:

```
Q = [];
for j = 1:n
  v = A(:,j);       % Take the jth original basis vector
  v = v-Q*(Q'*v);   % Make it orthogonal to q_i, i = 1:j-1
  v = v/norm(v);    % Normalize what remains
  Q = [Q, v];       % Append the result to the basis
end
```

Where does R appear in this algorithm? It appears thus:

```
Q = [];
R = zeros(m);
for j = 1:n
  v = A(:,j);      % Take the jth original basis vector
  rp = Q'*v;       % Project v onto previous basis vectors
  v   = v-Q*u;     % Make vector orthogonal to q_i, i = 1:j-1
  rjj = norm(v);   % Get the normalizing factor
  v   = v/rjj;     % Normalize what remains
  Q   = [Q, v];    % Append the result to the basis
  R(1:j,j) = [rp; rjj]; % ... and update R
end
```

That is, R accumulates the multipliers that we computed from the Gram-Schmidt procedure. This idea that the multipliers in an algorithm can be thought of as entries in a matrix should be familiar, since we encountered it before when we looked at Gaussian elimination.