Notes for 2016-10-03

1 Stability of QR

Last time, we discussed QR factorization via Householder reflectors or Givens rotations. It is not too difficult to show that applying a Givens rotations or Householder reflector to a matrix is backward-stable: if P is the desired transformation, the floating point result of PA is

$$\tilde{P}A = (P+E)A, \quad \|E\| \leq O(\epsilon_{\mathrm{mach}})\|A\|.$$

Moreover, orthogonal matrices are perfectly conditioned! Taking a product of j matrices is also fine; the result has backward error bounded by $jO(\epsilon_{\mathrm{mach}})\|A\|$. As a consequence, QR decomposition by Givens rotations or Householder transformations is ultimately backward stable.

The stability of orthogonal matrices in general makes them a marvelous building block for numerical linear algebra algorithms, and we will take advantage of this again when we discuss eigenvalue solvers.

2 Sparse QR

Just as was the case with LU, the QR decomposition admits a sparse variant. And, as with LU, sparsity of the matrix $A \in \mathbb{R}^{m \times n}$ alone is not enough to guarantee sparsity of the factorization! Hence, as with solving linear systems, our recommendation for solving sparse least squares problems varies depending on the actual sparse structure.

Recall that the R matrix in QR factorization is also the Cholesky factor of the Gram matrix: $G = A^T A = R^T R$. Hence, the sparsity of the R factor can be inferred from the sparsity of G using the ideas we talked about when discussing sparse Cholesky. If the rows of A correspond to experiments and columns correspond to factors, the nonzero structure of G is determined by which experiments share common factors: in general $g_{ij} \neq 0$ if any experiment involves both factors i and factor j. So a very sparse A matrix may nonetheless yield a completely dense G matrix. Of course, if R is dense, that is not the end of the world! Factoring a dense $n \times n$ matrix is pretty cheap for n in the hundreds or even up to a couple thousand, and solves with the resulting triangular factors are quite inexpensive.

If one forms Q at all, it is often better to work with Q as a product of (sparse) Householder reflectors rather than forming the elements of Q. One may also choose to use a "Q-less QR decomposition" in which the matrix Q is not kept in any explicit form; to form $Q^T b$ in this case, we would use the formulation $Q^T b = R^{-T} A^T b$.

As with linear solves, least squares solves can be "cleaned up" using iterative refinement. This is a good idea in particular when using Q-less QR. If $\tilde{A}^\dagger$ is an approximate least squares solve (e.g. via the slightly-unstable normal equations approach), iterative refinement looks like

$$\begin{aligned} r^k &= b - Ax^k \\ x^{k+1} &= x^k - \tilde{R}^{-1}(\tilde{R}^{-T}(A^T r_k)). \end{aligned}$$

This approach can be useful even when A is moderately large and dense; for example, $\tilde{R}$ might be computed from a (scaled) QR decomposition of a carefully selected subset of the rows of A.

3 Weighted least squares and company

So far, we have dealt primarily with the least squares problem with respect to the Euclidean norm associated with the standard inner product on $\mathbb{R}^n$. However, everything we have said works for other inner products as well. Let $\langle \cdot, \cdot \rangle_M$ be any inner product on $\mathbb{R}^n$, and let $\| \cdot \|_M^2$ be the associated inner product; then the problem

$$\text{minimize } \|Ax - b\|_M^2$$

yields the normal equations

$$A^T M r = 0.$$

We already saw one version of this when considering the problem of regression with normal errors drawn from a joint normal distribution; in that case, M was the inverse covariance matrix. But that is not the only place where the more general least squares picture is useful.

3.1 Least squares in polynomials spaces

For example, to optimally approximate a function on $[-1, 1]$ by a polynomial p of degree at most d, we would write

$$\forall q \in \mathcal{P}_d, \int_{-1}^{1} q(x)(p(x) - f(x))\, dx = 0.$$

If we were to express this in terms of the ordinary monomial basis, we could rewrite $p(x) = \sum_{j=0}^{d} c_j x^j$, and have

$$\int_{-1}^{1} x^i \left(\sum_j c_j x^j - f(x) \right) dx,$$

or, in matrix terms,

$$Gc = d$$

where $g_{ij} = \int_{-1}^{1} x^{i+j}\, dx$ and $d_i = \int_{-1}^{1} x^i f(x)\, dx$; here we have implicitly started indexing at zero. Another basis would yield a different matrix. In particular, if we take the Cholesky factorization $G = R^T R$ and define $U = R^{-1}$, then the polynomials

$$L_j(x) = \sum_{i=0}^{j} x^i u_{ij}$$

have the property that they form an orthonormal basis for the space $\mathcal{P}^d$ — essentially, this is the observation that $Q = AR^{-1}$ has orthonormal columns. The polynomials L_j are important enough that they have a name, the *normalized Legendre polynomials*; these polynomials play an important role in approximation theory and the theory of Gaussian quadrature. In fact, there is an explicit three-term recurrence for these polynomials which can be derived without resort to the Gram matrix; and we shall see this conneciton again when we talk later about Krylov subspace methods.

3.2 Weighting and re-weighting

What about a more prosaic case of re-weighting? For example, what if for some problem we care not about the residual, but the *relative residual* with components r_i/b_i? Then we seek to minimize

$$\sum_i (r_i/b_i)^2 = \|D^{-1}(Ax - b)\|^2$$

where $D = \text{diag}(b)$. Or, we might decide that we want to minimize

$$\sum_i w_i r_i^2$$

where the weights w_i are chosen to downweight outliers. How would we choose which equations count as outliers? A natural way to do this is to look at the residuals from an earlier fitting; this gives us the *iteratively reweighted least squares* (IRLS) algorithm.

4 Constrained case

Consider the weighted least squares problem

$$\text{minimize} \sum_{i=1}^m w_i r_i^2$$

where w_1 is much larger than the others. If we let $w_1 \to \infty$ while the others are fixed, what happens? We essentially say that we care about enforcing the first equation above all others, and in the limit we are solving the *constrained* least squares problem

$$\text{minimize} \sum_{i=2}^m w_i r_i^2 \text{ s.t. } r_1 = 0.$$

Unfortunately, if we actually try to compute this way, we are dancing on dangerous ground; as w_1 goes to infinity, so does the condition number of the least squares problem. But this is only an issue with the weighted formulation; we can formulate the constrained problem in other ways that are perfectly well-behaved.

In the remainder of this section, we address two ways of handling the linearly constrained least squares problem

$$\text{minimize } \|Ax - b\|^2 \text{ s.t. } C^T x = d,$$

by either eliminating variables (the *null-space method*) or adding variables (the method of *Lagrange multipliers*).

4.1 Null space method

In the null space method, we write an explicit expression for the solutions to $C^T x = d$ in the form $x^p + Wz$ where x^p is a particular solution to $C^T x^p = d$ and W is a basis for the null space of C^T. Perhaps the simplest particular solution is $x^p = (C^T)^\dagger d$, the solution with minimal norm; we can compute both this particular solution and an orthogonormal null space basis quickly using a full QR decomposition of C:

$$C = \begin{bmatrix} Q_1 & Q_2 \end{bmatrix} \begin{bmatrix} R_1 \\ 0 \end{bmatrix}, \quad x^p = Q_1 R_1^{-T} d, \quad W = Q_2.$$

Note that

$$C^T x^p = (R_1^T Q_1^T) x^p = d,$$

so this is indeed a particular solution. Having written an explicit parameterization for all solutions of the constraint equations, we can minimize the least squares objective with respect to the reduced set of variables

$$\text{minimize } \|A(x^p + Wz) - b\|^2 = \|(AW)z - (b - Ax^p)\|^2.$$

This new least squares problem involves a smaller set of variables (which is good); but in general, even if A is sparse, AW will not be. So it is appropriate to have a few more methods in our arsenal.

4.2 Lagrange multipliers

An alternate method is the method of *Lagrange multipliers*. This is an algebraic technique for adding equations to enforce constraints.

One way to approach the Lagrange multiplier method is to look at the equations for a constrained minimum. In order not to have a downhill direction, we require that the directional derivatives be zero in any direction consistent with the constraint; that is, we require $Cx = d$ and

$$\delta x^T A^T r = 0 \text{ when } C^T \delta x = 0.$$

The constraint says that admissible δx are orthogonal to the columns of C; the objective tells us the admissible δx should be orthogonal to the residual. So we need that $A^T r$ should lie in the column span of C; that is,

$$A^T r = -C\lambda$$

for some λ, and $Cx = d$. Putting this together, we have the KKT equations

$$\begin{bmatrix} A^T A & C \\ C^T & 0 \end{bmatrix} \begin{bmatrix} x \\ \lambda \end{bmatrix} = \begin{bmatrix} A^T b \\ d \end{bmatrix}.$$

These bordered normal equations are not the end point for constrained least squares with Lagrange multipliers, any more than the normal equations are the end point for unconstrained least squares. Rather, we can use this as a starting point for clever manipulations involving our favorite factorizations (QR and SVD) that reduce the bordered system to a more computationally convenient form.