Gary W. Howell

Solving Overdetermined Sparse Least Squares Problems by LU Preconditioning

512 Farmington Woods Dr
Cary
NC
27511
gwhowell@ncsu.edu
Marc Baboulin

We investigate how to use an LU factorization with the classical `lsqr` routine for solving overdetermined sparse least squares problems. Usually L is much better conditioned than A. Thus iterating with L instead of A results in faster convergence. Numerical experiments using Matlab illustrate the good behavior of our algorithm in terms of storage and convergence. This paper explores a preliminary shared memory implementation using SuperLU factorization.

As with other sparse linear systems, preconditioning techniques based on incomplete factorizations can improve convergence. One method to precondition the normal equations (**??**) is to perform an incomplete Cholesky decomposition of A^TA (e.g., RIF preconditioner [**?**]).

When A^TA and its Cholesky factorization are denser than A, it is natural to wonder if the LU factorization of A can be used in solving the least squares problem. In this paper we use an LU factorization of the rectangular matrix $A = \begin{pmatrix} A_1 \\ A_2 \end{pmatrix}$ where L is unit lower trapezoidal and U is upper triangular. For the nonpivoting case, the normal equations (**??**) become equation $\mathrm{L}^T Ly = c, (0)$

with $c = L^Tb, Ux = y$, and we can apply CG iterations on ().

Least squares solution using LU factorization has been explored by several authors. Peters and Wilkinson and Björck and Duff give direct methods. This work follows Björck and Yuan using conjugate gradient methods based on LU factorization, an approach worth revisiting because of the recent progress in sparse LU factorization. The lsqrLU algorithm presented here uses a lower trapezoidal L returned from a direct solver package. Here we use an $LU = PAQ$ factorization from shared memory SuperLU, comparing iteration with L to the $U^{-1}A$ iteration suggested by Saunders. Because several other direct solver packages offer scalable sparse LU factorizations, it appears likely that the algorithm used here can also be used to solve larger problems. The rate of linear conver-

gence for CG iterations on the normal equations is

$$K = \frac{\kappa - 1}{\kappa + 1},$$

where $\kappa = \sqrt{(A^T A)} = (A)$ and (A) denotes the 2-norm condition number of A (ratio of largest and smallest singular values of A). In our experiments, L is often much better conditioned than A, so convergence of the CG method is relatively rapid. Moreover, the total number of nonzeros in L and U is usually less than in the sparse Cholesky factorization of $A^T A$.