Exploiting sparsity in model matrices

Douglas Bates
Dept. of Statistics, University of Wisconsin - Madison, U.S.A.
E-mail: `bates@r-project.org`
and
Martin Mächler ETH - Zürich, Switzerland
E-Mail: `maechler@r-project.org`

Model matrices are used to define many types of statistical models and fitting such models continues to be a challenge because the models become more complex and are applied to ever larger data sets. Traditional dense matrix decomposition methods, such as the QR and Cholesky decompositions, perhaps augmented with accelerated BLAS or even GPU-based BLAS, continue to be effective when the model matrices have a moderate number of columns and it is only the number of rows that increases. However, in many applications the number of rows and the number of columns in the model matrix both increase for larger data sets. That is, we add "nuisance parameters" as we add observations. Many mixed-effects models have this property.

Storing and decomposing dense model matrices that expand in both height and width for larger data sets can quickly swamp computing resources. The computational burden becomes even greater if fitting the model requires iterative optimization of a criterion and the decomposition must be calculated or updated during each iteration. Exploiting sparsity in the model matrices, if present, can reduce both the amount of storage and the amount of computing required to fit the statistical model. Sparse matrix decomposition methods are particularly effective inside iterative algorithms because the decomposition is performed in a symbolic phase, which determines the location of the nonzeros in the result, followed by a numeric phase to determine the actual values in those positions. Frequently the symbolic phase is the most time-consuming but it only needs to be done once. Another, often overlooked, aspect of the sparse Cholesky decomposition (and, to a lesser extent, the sparse QR decomposition) is the use of fill-reducing permutations. Depending on the model, a reordering of the columns in the model matrix can greatly reduce the number of nonzeros in the decomposition and hence the amount of computing required to determine these values.

Sparse matrix methods implemented in the Matrix package for R are central to the methods for mixed models implemented in the lme4 package. We will describe in detail the theory and computational methods for linear, generalized linear and nonlinear mixed models as implemented in lme4 and how access to the CHOLMOD library of C functions for the sparse Cholesky decomposition allows for a particularly efficient mixed model implementation.

One impediment to more widespread use of sparse matrix methods in R is the inability to directly produce a sparse model matrix from the model.matrix function. In lme4 sparse model matrices are used only for the random effects and the nature of the random effects terms makes it feasible to have custom code to produce the sparse matrices directly. It may be worthwhile providing a more general mechanism through the model.matrix function.