Chapter 1
Solving Some Structured Linear Least-Squares Problems

Linda Kaufman*

Abstract

Linear least-squares problems involving very large, completely dense matrices with a highly special structure arise in two applications—system identification and separable nonlinear least-squares with multiple data sets. Generic dense methods applied to such problems would be inordinately costly in terms of both speed and storage. We describe various techniques for exploiting the structure that differ in operation count, storage requirements, accuracy, and suitability for parallel computation.

1 Introduction

In this paper we consider solving a class of linear least-square problems, minimizing $||Ax - b||_2$ in which the least-squares matrix A has a particular structure. In a typical problem in this class the matrix A is "long and skinny", e.g. 100,000 by 200, dense, and it is not known a priori whether elements of A are repeated in a particular pattern. However, the data needed to generate A can be stored in much less space than the number of nonzeroes in A. Moreover, during a direct decomposition of A, the same row transformation can often be applied simultaneously to different rows to annihilate different nonzero elements. Although our structured matrix is rather specific, we hope the techniques to be described can be used in other applications.

Given an $m \times n$ matrix G and an $r \times n$ matrix H, our special least-squares matrix A has the form

$$A = \begin{pmatrix} GH^{(1)} \\ \vdots \\ GH^{(r)} \end{pmatrix}, \quad \text{where} \quad H^{(i)} = \begin{pmatrix} h_{i1} & & \\ & \ddots & \\ & & h_{in} \end{pmatrix}. \tag{1}$$

In Section 2 of this paper we give three instances where matrices of the first form in (1) or variants have arisen. In Section 3 we discuss how the matrix A^TA can be easily obtained for A given in (1) and for some generalizations of (1) that come from the multiple data set separable nonlinear least-squares problem. In Section 4 we talk about obtaining the QR decomposition of A.

2 Problem Origins

In this section we give three examples which, in their simplest versions, involve a matrix with the form (1).

The first example is the multiple data set separable nonlinear least-squares problem (MDSN). In separable nonlinear least-squares problems for given data b, one wishes to find

*AT&T Bell Laboratories, Murray Hill, New Jersey 07974 (lck@research.att.com).

the vectors x and h which minimize $||\Phi(x)h - b||_2$. For example one might wish to fit data to models of the form

$$h_1 e^{tx_1} + h_2 e^{tx_2}, \tag{2}$$

$$h_1 e^{x_1(t-x_2)} + h_2 e^{x_3(t-x_4)}, \quad \text{or} \tag{3}$$

$$h_1 e^{x_1(t-x_2)} + h_2 e^{x_3(t-x_2)}. \tag{4}$$

In the multiple data set case, r sets of the data b are given and one wishes to find r sets of the linear parameters but one common set of the nonlinear parameters x. For this problem, where the number of linear parameters far exceeds the number of nonlinear parameters, the approach suggested by Golub and Pereyra [GP73] of minimizing $\| (\Phi(x)\Phi^+(x) - I)b \|_2^2$ (an equivalent problem) is attractive.

Most nonlinear least-squares solvers are iterative algorithms in which each iteration involves solving a linear least-squares problem with the Jacobian A. If there are r data sets, then the Jacobian may be partitioned as

$$A = \begin{pmatrix} A^{(1)} \\ \vdots \\ A^{(r)} \end{pmatrix}.$$

As shown in Kaufman and Sylvester [KS92], the k^{th} column of $A^{(i)}$ has the form

$$a_k^{(i)} = G^{(k)} h^{(i)}$$

where if x_k appears in term p in the model, then the p^{th} column of $G^{(k)}$ will be nonzero and $h^{(i)}$ is the current approximation to the linear parameters of the i^{th} data set. If the i^{th} nonlinear variable appears only in the i^{th} term in the model, as in (2), then we say we have a matched model. For a matched model if one forms the matrix G from the nonzero columns of all the $G^{(k)}$'s and generates a matrix H whose i^{th} row is $h^{(i)^T}$, then the Jacobian has exactly the form of A in (1).

The second problem is a particular nonlinear least-squares problem that arises in a frequency domain approach to a system identification problem where one wants to estimate modal parameters from multiple-driver, multiple-receiver transfer function data. In this problem for given data b and γ, one wishes to determine the c', σ's and h's which minimize

$$\sum_{i=1}^{r} \sum_{j=1}^{m} \sum_{s=1}^{S} \Big(\sum_{k=1}^{n} c_{sk} h_{ik} \ /(\gamma_j - \sigma_k) - b_{ijs} \Big)^2. \tag{5}$$

If one forms the matrix G where $g_{jk} = 1/(\gamma_j - \sigma_k)$, then (5) becomes a standard separable nonlinear least-squares problem with the matrix $\Phi = I \otimes A$ where the matrix A has the form (1). The application of the Golub-Pereyra idea to this problem requires the QR decomposition of Φ, which requires the QR decomposition of A.

The third example again arises when solving a system identification problem. In this problem one is given two complex $r \times n$ matrices G and J, an $m \times n$ complex matrix H and a complex right-hand side b and one wants to find the complex n-vector z that solves the linear least-squares problem

$$\text{minimize} \quad \sum_{i=1}^{m} \sum_{j=1}^{r} \Big(\Big(\sum_{k=1}^{n} G_{ik} H_{jk} z_k + J_{ik} H_{jk}^* z_k^* \Big) - b_{ij} \Big)^2, \tag{6}$$

where the superscript "$*$" denotes the complex conjugate. If one neglects the term with J and considers all matrices as real, then (6) becomes a standard least squares problem of minimizing $\|Ax - b\|_2^2$, where A has the form given in (1). The solution of this least-squares problem, including the treatment of the term with J and the complex arithmetic, is given in [KSW93].

3 The normal equations

The most obvious method for solving the linear least-squares problem of minimizing $\|Ax - b\|_2$ is to use the classical *normal equations* $A^TAx = A^Tb$. We know from the special structure of A shown in (1) that

$$A^TA = H^{(1)}G^TGH^{(1)} + H^{(2)}G^TGH^{(2)} + \cdots H^{(r)}G^TGH^{(r)}. \tag{7}$$

It follows from the definition of $H^{(i)}$ in (1) that the (i,j) element of A^TA is given by

$$(A^TA)_{ij} = (G^TG)_{ij}\,(H^TH)_{ij}. \tag{8}$$

The matrix A^TA can consequently be obtained by separately forming the two explicit $n \times n$ matrices G^TG and H^TH, and then multiplying together their (i,j) elements, using symmetry in the obvious way. The total work required to form A^TA using (7) is approximately $\frac{1}{2}(m+r)n^2$ multiplications. For the typical values $m = 500$ and $r = 200$ for (6) in [KSW93], forming A^TA using the structure of A requires about $350n^2$ multiplications, compared to $50,000n^2$ multiplications if structure is not exploited.

It is possible to obtain formulae for the normal equations approach using the Jacobian in the MDSN problem for a non-matched model. The simplest cases are similar to (3), where each nonlinear variable appears only once. If variable x_i appears only in term η_i, then

$$(A^TA)_{ij} = (G^TG)_{ij}(H^TH)_{\eta_i,\eta_j}.$$

If, however, some x_i appears more than once in the model, then $G^{(i)}$ will have more than one nonzero column. Following the suggestion in [KS92], let G represent the concatenation of all the nonzero columns of the $G^{(i)}$'s. Let ν_k denote which $G^{(i)}$ the k^{th} column of G came from and let η_k denote which column of $G^{(i)}$ it was from. Assuming there are s columns in G, the following algorithm will determine A^TA :

Set $A^TA \leftarrow 0$
For $k = 1, ..., s$
  For $j = 1, ..., s$
    Set $(A^TA)_{\nu_k,\nu_j} \leftarrow (A^TA)_{\nu_k,\nu_j} + (G^TG)_{k,j}(H^TH)_{\eta_k,\eta_j}$

For example, for (4), A would have 3 columns, G would have 4 columns, since x_2 appears twice, and H would have 2 columns, and

$$\begin{aligned}
(A^TA)_{11} &= (G^TG)_{11}(H^TH)_{11}, \quad (A^TA)_{21} = (G^TG)_{21}(H^TH)_{11} + (G^TG)_{31}(H^TH)_{21},\\
(A^TA)_{22} &= (G^TG)_{22}(H^TH)_{11} + 2(G^TG)_{23}(H^TH)_{21} + (G^TG)_{33}(H^TH)_{22},\\
(A^TA)_{31} &= (G^TG)_{41}(H^TH)_{21}, \quad (A^TA)_{32} = (G^TG)_{24}(H^TH)_{21} + (G^TG)_{34}(H^TH)_{22}, \quad \text{and}\\
(A^TA)_{33} &= (G^TG)_{44}(H^TH)_{22}.
\end{aligned}$$

4 The QR decomposition

In [KSW93] several algorithms for the QR decomposition of A with the form(1) are explained in detail. They all begin with obtaining the QR decomposition of G

$$QG = \tilde{R} = \begin{pmatrix} R \\ 0 \end{pmatrix}, \tag{9}$$

where Q is an orthogonal matrix and R is an upper triangular matrix. The matrix $\tilde{A}$ is then formed by deleting the zero rows of the matrix $(I \otimes Q)A$. The matrix $\tilde{A}$ will have nr rows and will look like r upper triangular matrices stacked one on top of each other as in the following figure.

$$\tilde{A} = \begin{pmatrix} \times & \times & \times \\ & \times & \times \\ & & \times \\ \times & \times & \times \\ & \times & \times \\ & & \times \\ \times & \times & \times \\ & \times & \times \\ & & \times \\ \times & \times & \times \\ & \times & \times \\ & & \times \end{pmatrix}.$$

When $n \geq r/2$, the QR decomposition of A can be obtained by reducing $\tilde{A}$ to upper triangular form. When $n < r/2$, it is more economical to first permute the rows of $\tilde{A}$ so that the first row of each of the r triangles is moved into the first block, the second rows of the triangles are moved into the second block, etc. Call the resulting matrix $\hat{A}$. The i^{th} block of $\hat{A}$, $\hat{A}^{(i)}$, will be $HR^{(i)}$ where $R^{(i)}$ is a diagonal matrix whose diagonal elements are the i^{th} row of R. Now form the QL decomposition of H

$$VH = \tilde{L} = \begin{pmatrix} L \\ 0 \end{pmatrix}, \tag{10}$$

where V is orthogonal and L is lower triangular. Let $\breve{A}$ represent the nonzero rows of the matrix $(I \otimes V)\hat{A}$. When $n = 5$, for example, $\breve{A}$ has the following form.

$$\breve{A} = \begin{pmatrix} \times & & & & \\ \times & \times & & & \\ \times & \times & \times & & \\ \times & \times & \times & \times & \\ \times & \times & \times & \times & \times \\ & \times & & & \\ & \times & \times & & \\ & \times & \times & \times & \\ & \times & \times & \times & \times \\ & & \times & & \\ & & \times & \times & \\ & & \times & \times & \times \\ & & & \times & \\ & & & \times & \times \\ & & & & \times \end{pmatrix}.$$

Finally $\breve{A}$ should be reduced to triangular form(see[KSW93]). Thus the total algorithm for obtaining the QR decomposition of A can be written as follows:

1. Determine the decompositions 9 and 10.
2. Set $\breve{A}$ to the nonzero rows of $(V \otimes I)(I \otimes Q)A$
3. Reduce $\breve{A}$ to triangular form.

When $m = 500, r = 200$, and $n = 60$, by using the structure of A as in the above algorithm the cost of obtaining the QR decomposition of A is reduced from 3.6×10^8 operations to 3.5×10^6 operations.

For the MDSN problem on a non-matched example, the matrices G and H and vectors η and ν of Section 3 can be used to derive $\breve{A}$. Assuming R in (9) is derived from G with s columns and L in (10) from H with l columns, then one can proceed as follows:

Set $\breve{A} \leftarrow 0, m \leftarrow 1$
For $i = 1, ..., s$
    For $j = 1, ..., l$
        For $k = i, ..., s$
            If $\eta_k > j$ then set $\breve{A}_{m,\nu_k} \leftarrow \breve{A}_{m,\nu_k} + R_{i,k} L_{j,\eta_k}$
        If the m^{th}row of $\breve{A}$ is nonzero, increment m

In the worst case, $\breve{A}$ will have $l \times s$ nonzero rows. The exact number of nonzero rows will depend on the ordering of the variables. For (4) the matrix $\breve{A}$ would look like

$$\breve{A} = \begin{pmatrix} \times & \times & \\ \times & \times & \times \\ & \times & \\ & \times & \times \\ & \times & \times \\ & & \times \end{pmatrix}.$$

Of course much of the work in solving least-squares problem entails applying the transformations to the right-hand side. If one partitions the right-hand side according to the blocks of A and imagines each partition as a column in a matrix B, then applying $(I \otimes Q)$ to the right-hand side is like applying Q on the left to B and applying $(V \otimes I)$ to the right-hand side is like applying V on the right to B (i.e. like applying column transformations to B).

We have shown that exploiting the structure of certain large dense linear least-squares problems leads to substantial improvements compared to using off-the-shelf "black-box" techniques. These matrices arise in a system identification problem and in separable nonlinear least-squares problems with multiple data sets. We anticipate that similar approaches will be developed for other problems.

References

[GP73] G. H. GOLUB AND V. PEREYRA, *The differentiation of pseudo-inverses and nonlinear least-squares problems whose variables separate*, SIAM J. Numer. Anal., 10 (1973), pp. 413–432.

[KS92] L. KAUFMAN AND G. SYLVESTER, *Separable nonlinear least squares with multiple right-hand sides*, SIAM J. Mat. Anal. Appl., 13 (1992), pp. 68–89.

[KSW93] L. KAUFMAN, G. S. SYLVESTER, AND M. H. WRIGHT, Structured linear least-squares problems in system identification and separable nonlinear data fitting, Numerical Analysis Manuscript 93–14, AT&T Bell Laboratories, Murray Hill, New Jersey, 1993.