International Mathematical Forum, Vol. 7, 2012, no. 52, 2549 - 2554

An Application of the Dulmage-Mendelsohn Decomposition to Sparse Null Space Bases of Full Row Rank Matrices

Mostafa Khorramizadeh

Department of Mathematical Sciences
Shiraz University of Technology
Shiraz 71555-313, Iran
m.khorrami@sutech.ac.ir

Abstract. In this paper we present an efficient algorithm for computing a sparse null space basis for a full row rank matrix. We first apply the ideas of the Markowitz's pivot selection criterion to a rank reducing algorithm to propose an efficient algorithm for computing sparse null space bases of full row rank matrices. We then describe how we can use the Dulmage-Mendelsohn decomposition to make the resulting algorithm more efficient.

1. Introduction

Let $A = (a_1, \cdots, a_m)^T \in R^{m\times n}$, $m < n$. The set of all $x \in R^n$, satisfying $a_t^T x = 0$, $1 \leq t \leq m$, is called the null space of A. If every vector in the null space of A can be written as a linear combinations of the columns of N, then N is called a null space generator of A. If the columns of N, a generator of A, are linearly independent, then N is called a null space basis of A. The sparse null space basis problem (*SNBP*) is to find a basis with fewest nonzeros for the null space of a sparse matrix A. *SNBP*s appear in various branches of mathematics, engineering and computer science, and its effective solution is a key element for the success of various algorithms such as constrained nonlinear programming algorithms [1], some special interior point algorithms for optimization problems [10], the dual variable method for solving the Navier-Stokes equations [8] and force methods for structural optimization [9].

To explain our ideas, at first we need to briefly describe a rank reducing algorithm for computing a null space basis and the Markowitz's pivot selection criterion.

1.1. Rank reducing algorithm.

Assume that $A = (a_1, \cdots, a_m)^T \in R^{m\times n}$, $m < n$, has full row rank. By definition

$$N(a_1^T) = \{y \in R^n : a_1^T y = 0\}$$

and the orthogonal complement of $N(a_1^T)$ is $R(a_1) = \{\alpha a_1 : \alpha \in R\}$. Therefore, if $G_1 \in R^{n-1\times n}$ satisfies

$$G_1 y = 0 \Leftrightarrow y \in R(a_1)$$

for every $y \in R^n$, then, G_1 is a basis for $N(a_1^T)$. For $1 \leq i \leq m$, let $A_i = (a_1, \ldots, a_i)$ and $s_i = M_i a_i$, where $M_i^T \in R^{n\times(n-i+1)}$ is a basis for $N(A_{i-1}^T)$. Moreover, let $G_i \in R^{(n-i)\times(n-i+1)}$ satisfies

$$G_i y = 0 \Leftrightarrow y \in R(s_i),$$

for every $y \in R^{n-i+1}$, and $M_{i+1} = G_i M_i \in R^{(n-i)\times n}$. In the following we show that M_{i+1}^T is a basis for $N(A_i^T)$. Since A has full row rank, we have $dim(N(A_i^T)) = n - i$. So it is sufficient to show that the columns of M_{i+1}^T generate $N(A_i^T)$. Indeed, let $x \in N(A_i^T)$, then since $x \in N(A_{i-1}^T)$ there exists some $z \in R^{n-i+1}$ so that $x = M_i^T z$. Moreover, we have

$$0 = a_i^T x = a_i^T M_i^T z = s_i^T z.$$

Since $R^{n+i-1} = R(G_i^T) \oplus N(G_i)$, we can write $z = z_R + z_N$, where, $z_R \in R(G_i^T)$ and $z_N \in N(G_i)$. In the following we will show that $z_N = 0$. By definition there exists some $y_R \in R^{n-i}$ so that $z_R = G_i^T y_R$. If $z_N \neq 0$ then since $G_i z_N = 0$, there exists some $\alpha \neq 0$ so that $z_N = \alpha s_i$. Moreover, we have $s_i^T z = \alpha s_i^T s_i + s_i^T G_i^T y_R$. Since A has full row rank we have $s_i \neq 0$. By definition of G_i we have $G_i s_i = 0$. Therefore, $s_i^T z = \alpha \|s_i\|^2 \neq 0$. This contradiction shows that $z_N = 0$ and hence $z = z_R = G_i^T y_R$. This shows that $x = M_i^T z = M_i^T G_i^T z_R = M_{i+1}^T z_R$ and therefore, M_{i+1}^T is a basis for $N(A_i^T)$. The above considerations suggests the following rank reducing algorithm for computing a null space basis of A.

Algorithm 1. *Rank reducing algorithm for computing null space basis.*

Step 1: *Let $M_1 \in R^{n\times n}$ be the identity matrix in $R^{n\times n}$. Set $i = 1$.*
Step 2: *Compute $s_i = M_i a_i$.*
Step 3: *Compute $M_{i+1} = G_i M_i$, where $G_i \in R^{(n-i)\times(n-i+1)}$ is such that we have $G_i y = 0$ if and only if $y = \alpha s_i$, for some $\alpha \in R$.*
Step 5: ***If*** *$i = m$* ***then stop*** *(M_{m+1}^T is the null space basis for A)* ***else*** *let $i = i + 1$ and* ***go to*** **Step 2**.

1.2. **Markowitz's pivot selection criterion.** Here, we describe Markowitz's pivot selection criterion [11]. Let r_i^t and c_j^i be the number of nonzero elements in row t and column j of the remaining $(n-i)\times(n-i)$ matrix after an application of i iterations of the Gaussian elimination to the matrix A. The Markowitz pivot selection criterion is a local greedy strategy that selects from the remaining submatrix a nonzero element a_{tj}^i that corresponds to minimum Markowitz count, $(r_t^i - 1)(c_j^i - 1)$. In practice the minimum is taken over all entries satisfying the inequality,

$$|a_{tj}^i| \geq u\, \mathtt{max}\{|a_{tl}^i|, \quad l \geq i\}, \tag{1}$$

or the inequality,

$$(2) \qquad |a^t_{tj}| \geq u\, \texttt{max}\{|a^i_{lj}|, \quad l \geq i\},$$

where u, $0 < u \leq 1$, is a constant. The element a^i_{tj} is forced to satisfy (1) or (2) to insure the numerical stability of the algorithm as well. This is done to minimize the number of fill–ins in the next iteration [5, 6].

Here, we first apply a similar criterion for the selection of the G_i in rank reducing algorithm to preserve sparsity. Moreover, we consider the matrix $\widetilde{A}_i = (M_i a_{i+1}, \cdots, M_i a_m)^T$ as the remaining matrix of the ith iteration and then apply the Markowitz's pivot selection criterion along with inequality (1) or (2) to $\widetilde{A}_i$ to choose G_i effectively. The resulting algorithm preserves sparsity and since in every iteration we only need to compute a sparse matrix–vector and a sparse matrix–matrix product, the resulting algorithm generate the sparse null basis effectively. Then, we use the Delmuge–Mendelsohn decomposition to make our proposed algorithm more efficient. Indeed, we first apply this decomposition to the full row rank matrix A to obtain a sparse submatrix of A whose null space basis completely determine a sparse null space basis of A. Then, we apply the sparse rank reducing algorithm to the resulting submatrix. Finally, we examine the numerical performance of our proposed algorithm and justify its efficiency.

In section 2, we describe the application of the Markowitz pivot selection criterion to the rank reducing algorithm to preserve sparsity. In section 3, we explain how the Dulmege–Mendelsohn decomposition can be used to make our proposed algorithm more efficient. In section 4, we consider the numerical performance of our proposed algorithm and justify its efficiency.

2. **Sparse rank reducing algorithm**

Here, we propose an effective algorithm for the *SNBP*. We intend to compute a sparse null space basis for the matrix $A \in R^{m\times n}$. In the beginning of the algorithm, we let M_1 be the identity matrix in $R^{n\times n}$. Suppose that we are at the ith iteration of the rank reducing algorithm and let h_k^T, $1 \leq k \leq n-i+1$, be the kth row of M_i, $A_{m-i} = (a_i, \cdots, a_m)^T$ and $\widetilde{A}_i \in R^{(m-i+1)\times(n-i+1)}$ be given by

$$\widetilde{A}_i = A_{m-i} M_i^T = \begin{pmatrix} a_i^T M_i^T \\ \vdots \\ a_m^T M_i^T \end{pmatrix} = \begin{pmatrix} A_{m-i}h_1 & \cdots & A_{m-i}h_{n-i+1} \end{pmatrix}.$$

Similar to the Markowitz's pivot selection criterion, one may think of the selection of the parameters of the rank reducing algorithm, corresponding to a minimal product $(\widetilde{r}^i_t - 1)(\widetilde{c}^i_j - 1)$, over all entries satisfying the inequality,

$$(3) \qquad |\widetilde{a}^i_{tj}| \geq u\, \texttt{max}\{|\widetilde{a}^i_{tl}|, \quad l \geq i\},$$

where u, $0 < u \leq 1$, is a constant, $\widetilde{r}^i_t$ and $\widetilde{c}^i_j$ are the number of nonzero elements in row t and column j of $\widetilde{A}_i$, respectively, and $\widetilde{a}^i_{tj}$ denotes the element in the tth row and jth column of $\widetilde{A}_i$. Moreover, let $\widetilde{a}^i_{t_i j_i}$ be the element, corresponding to a minimal product $(\widetilde{r}^i_t - 1)(\widetilde{c}^i_j - 1)$, over all entries satisfying the inequality (3), and then at the ith iteration of the rank reducing algorithm, let $s_i = M_i a_{t_i}$ and set

$$G_i = \begin{pmatrix} 1 & & & -s_i^1/s_i^{j_i} & & & \\ & \ddots & & \vdots & & & \\ & & 1 & -s_i^{j_i-1}/s_i^{j_i} & & & \\ & & & -s_i^{j_i+1}/s_i^{j_i} & 1 & & \\ & & & \vdots & & \ddots & \\ & & & -s_i^{n-i+1}/s_i^{j_i} & & & 1 \end{pmatrix}.$$

Note that the matrix G_i is an $(n-i) \times (n-i+1)$ matrix, obtained by adjoining the vector

$$s'_i = (-s_i^1/s_i^{j_i}, \cdots, -s_i^{j_i-1}/s_i^{j_i}, -s_i^{j_i+1}/s_i^{j_i}, \cdots, -s_i^{n-i+1}/s_i^{j_i})^T, \tag{4}$$

as a new kth column of the identity matrix in $R^{(n-i)\times(n-i)}$. By performing simple algebraic multiplications, we can verify that for every y, $G_i y = 0$ if and only if $y = \alpha s_i$, for some scalar $\alpha \in R$. Since the computation of the matrix $\widetilde{A}_i$ in every iteration of the algorithm is costly and time consuming, we choose t_i and j_i so that $\widetilde{r}^i_{t_i}$, $\widetilde{c}^i_{j_i}$ and hence the product $(\widetilde{r}^i_{t_i} - 1)(\widetilde{c}^i_{j_i} - 1)$ are expected to be small. Since the tth row of $\widetilde{A}_i$ is $a_t^T H_i^T$ and $\widetilde{r}^i_t$ is the number of nonzeros of $a_t^T H_i^T$, we let t_i be the index of the row of A that, among all rows of A not considered so far, has the minimum number of nonzeros and compute $s_i = M_i a_{t_i}$. Similarly, since the jth column of $\widetilde{A}_i$ is $A_{m-i} h_j$, we let j_i be the index of the column of $\widetilde{A}_i$ that, among all nonzero elements of s_i satisfying (3), corresponds to the row of M_i with a minimal number of nonzeros.

Considering the above argument, to determine the matrix G_i, we first let a_{t_i} be the row of A that, among all rows of A not considered so far, has a minimal number of nonzeros. Then, we compute the vector $s_i = M_i a_{t_i}$ and let j_i, among all nonzero elements of s_i that satisfy (3), correspond to the row of M_i with a minimal number of nonzeros. Finally, we let G_i be the adjoined identity matrix in $R^{(n-i)\times(n-i)}$, with s'_i as in (4) added as a new j_ith column. Our proposed algorithm is expected to generate a sparse null space basis. The resulting sparse rank reducing algorithm follows next.

Algorithm 2. *SRRP (Sparse Rank Reducing algorithm)*

Step 1: *Let $a_{t_1}^T, \cdots, a_{t_m}^T$ be the rows of A in ascending order with respect to their number of nonzero elements. Let M_1 be the identity matrix in $R^{n\times n}$ and $u \in [0, 1]$. Set $i = 1$.*

Step 2: *Compute $s_i = M_i a_{t_i}$.*

Step 3: *Let* $s_i^{j_i}$ *be a nonzero element of* s_i*, which, among all nonzero elements of* s_i*, satisfying*

$$|s_i^{j_i}| \geq u \, \textbf{\textit{max}}\{|s_i^k|, \quad 1 \leq k \leq n-i+1\}, \tag{5}$$

corresponds to the row of M_i *with a minimal number of nonzeros (select the first one if there are multiple minima).*

Step 5: *Let*

$$s_i = (s_i^1, \cdots, s_i^{j_i-1}, s_i^{j_i}, s_i^{j_i+1}, \cdots, s_i^{n-i+1}), \quad s_i^{j_i} \neq 0,$$

$$G_i = \begin{pmatrix} 1 & & & -s_i^1/s_i^{j_i} & & & \\ & \ddots & & \vdots & & & \\ & & 1 & -s_i^{j_i-1}/s_i^{j_i} & & & \\ & & & -s_i^{j_i+1}/s_i^{j_i} & 1 & & \\ & & & \vdots & & \ddots & \\ & & & -s_i^{n-i+1}/s_i^{j_i} & & & 1 \end{pmatrix}.$$

and compute $M_{i+1} = G_i M_i$.

Step 6: ***If*** $i = m$ ***then*** ***stop*** *(*M_{m+1} *is a basis for the null space of* A*)* ***else*** *let* $i = i+1$ *and* ***go to*** **Step 2**.

3. Making the algorithm more efficient

In this section we explain how we can improve the efficiency of *SRRP* using Delmuge–Mendelsohn decomposition.

Suppose that we can permute the columns and rows of A, so that the resulting permuted matrix has in the following form:

$$A' = \begin{pmatrix} B & \mathbf{0}_{r\times n-r} \\ C & H \end{pmatrix}$$

where, $B \in R^{r\times r}$, $C \in R^{m-r\times r}$, $H \in R^{m-r\times n-r}$ and $\mathbf{0}_{r\times n-r}$ is the zero matrix in $R^{r\times n-r}$. It can be easily verified that $rank(N(H)) = n - r$. Moreover, if $y \in N(H)$, then $\begin{pmatrix} \mathbf{0}_{r\times 1} \\ y \end{pmatrix} \in N(A')$. Therefore, we can obtain a basis for $N(A')(= N(A))$ by computing a basis for $N(H)$. The above consideration can be utilized to improve the efficiency of *SRRP*. Indeed, we can apply the *SRRP* to H instead of A. If A has full row rank, then the Delmuge–Mendelsohn decomposition [2, 3, 4] provide us with a permutation of the columns and rows of A so that the resulting permuted matrix has the form of A'. Therefore, to improve the efficiency of *SRRP* we first compute the Delmuge–Mendelsohn decomposition and then apply the *SRRP* on a submatrix of A and compute the sparse null space of A as described above.

4. Concluding remarks

In this paper we described an efficient algorithm for sparse null space basis problem and then explained how we can improve the efficiency of our proposed algorithm by using the Dulmage-Mendelsohn decomposition.

Acknowledgment: The author thank the Research Council of Shiraz University of Technology for its support.

References

[1] T.S. Coleman and J.J. More, Estimation of sparse Jacobian matrices and graph coloring problem. *SIAM J. Num. Anal.* **20**, 187–209 (1983)

[2] A. L. Dulmage and N. S. Mendelsohn, Coverinfgs of bipartite graphs. *Canadian Journal of Mathematics* **10**, 517–534 (1958)

[3] A. L. Dulmage and N. S. Mendelsohn, A structure theory of bipartite graphs of finite exterior dimension. *Transactions of the royal society of canada section III* **53**, 1–13 (1959)

[4] A. L. Dulmage and N. S. Mendelsohn, Two algorithms of bipartite graphs. *Journal of the society of industrial and applied mathematics* **11**, 183–194 (1963)

[5] T. A. Davis, Algorithm 832: UMFPACK, an unsymmetric-pattern multifrontal method. *ACM Trans. Math. Softw.* **30(2)**, 196–199 (2004)

[6] T.A. Davis, Direct methods for sparse linear systems. SIAM, Philadelphia, (2006)

[7] J.R. Gilbert and M.T. Heath, Computing a sparse basis for the null space. *SIAM Journal on Algebraic and Discrete Methods* **8(3)**, 446–459 (1987)

[8] C. Hall, Numerical solution of Navier-Stokes problems by the dual variable method. *SIAM Journal on Algebraic and Discrete Methods* **6**, 220–236 (1985)

[9] M.T. Heath, R.J. Plemmons and R.C. Ward, Sparse orthogonal schemes for structural optimization using the force method. *SIAM J. Sci. Sta. Comput.* **5**, 514–532 (1984)

[10] K. Kim and J.L. Nazareth, A primal null-space affine-scaling method. *ACM Transactions on Mathematical Software* **20**, 373 - 392 (1994)

[11] H. M. Markowitz, The elimination form of the inverse and its application to linear programming. *Management Sci.* **3**, 255-269 (1957)

Received: May, 2012