ACTA UNIVERSITATIS APULENSIS No 11/2006

Proceedings of the *International Conference on Theory and Application of Mathematics and Informatics ICTAMI 2005* - Alba Iulia, Romania

ALGORITHM TO WORKING WITH SPARSE MATRICES

Vlad Monescu

Abstract. A new method to memorizing a sparse matrix is developed here. Blocks of matrices are binary converted and take into consideration when the elements have to be accessed. The algorithm is compared with some classic ones and computational results are presented.

2000 *Mathematics Subject Classification*: 65F50, 65F30, 68Q25.

1. Introduction

The ideea to take in consideration the large number of zeros of a matrix and their location was initiated in the second half of nineteenth century by electrical engineers. A $n \times m$ matrix is a sparse matrix if the number of nonzero entries is much smaller than $n \times m$. There are two problems: how to retain in minimum memory space a sparse matrix and how to access it's elements. Sparse matrices arise in optimization problems, solutions to partial differential equations, structural and circuit analysis and computational fluid dynamics. Sparse matrices can be huge; dimensions on the order of 100,000 are not uncommon. Only by exploiting sparsity can we hope to be able to manipulate such a matrix on a computer.

2. The algorithm

We propose a method to memorizing a sparse matrix. For this let A be a sparse matrix, $A \in M_{n,m}(\mathbf{R})$. For the following construction it will be chosen the numbers $p, q \in \mathbf{N}^*$. The rows of matrix are divided in groups of p rows and the columns are divided in groups of q columns. If $n \bmod p \neq 0$ or $m \bmod q \neq 0$ then supplementary rows, respectively columns are added containing null values. The matrix we obtain by adding rows or columns is equivalent with the initial matrix. Each of $([\frac{n-1}{p}]+1) \times ([\frac{m-1}{q}]+1)$ blocks of elements is $p \times q$ binary converted. All positions containing nonzero values are considered 1. Thus we obtain a new matrix T which have nonnegative integer elements as result of conversion in $p \times q$ bits in a matricial disposal.

If we denote N the number of nonzero elements in A then $N + ([\frac{n-1}{p}] + 1) \times ([\frac{m-1}{q}] + 1)$ memory locations are needed to memorize the sparse matrix A using this method.

Example.

For the matrix $A = \begin{pmatrix} 11 & 0 & 0 & 14 & 5 & 0 & 0 & 8 \\ 0 & 0 & 0 & 5 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 6 & 0 & 0 & 0 & 0 \\ 4 & 0 & 0 & 7 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 4 \\ 0 & 0 & 0 & 0 & 10 & 0 & 0 & 0 \\ 17 & 0 & 0 & 0 & 0 & 0 & 58 & 0 \end{pmatrix}$, if $p = 4$ and $q = 4$ we have the following configuration

$$A' = \begin{pmatrix} 1 & 0 & 0 & 1 & \cdot & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & \cdot & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & \cdot & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & \cdot & 0 & 1 & 0 & 0 \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ 0 & 0 & 0 & 0 & \cdot & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & \cdot & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & \cdot & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & \cdot & 0 & 0 & 0 & 0 \end{pmatrix}.$$

It has been added a row with 0. Each group of 16 bits represents a non-negative integer number in binary conversion. Thus we obtain

$$T = \begin{pmatrix} 37145 & 37124 \\ 128 & 6176 \end{pmatrix}.$$

The vector of nonzero elements (ordered by their appeareances in their blocks) is

$$w^T = \begin{pmatrix} 11 & 14 & 5 & 6 & 4 & 7 & 5 & 8 & 1 & 1 & 17 & 4 & 10 & 58 \end{pmatrix}.$$

Once we have T, and w it is necessary to access the elements of matrix A. Let it be $v, b, k \in \mathbf{N}$. We denote

- $N_b(v)$ - number of nonzero bits of v in the b-bits binary representation,
- $P_{b,k}(v)$ - position of the k-th nonzero bit of v in the b-bits binary representation,

- $B_{b,k}(v)$ - value of the k-th bit of v in the b-bits binary representation.

If we have to access the element $a_{i,j}$ and

$$B_{b,pq-(i \ mod \ p)q-j \ mod \ q}(t_{[\frac{i-1}{p}]+1,[\frac{j-1}{q}]+1}) = 0$$

then $a_{i,j} = 0$, else the position s of the element $a_{i,j}$ in w is given by

$$s = \sum_{k=1}^{[\frac{i-1}{p}]} \sum_{l=1}^{[\frac{m-1}{q}]+1} N_{pq}(t_{k,l}) + \sum_{l=1}^{[\frac{j-1}{q}]} N_{pq}(t_{[\frac{i-1}{p}]+1,l}) - N_{(i \ mod \ p)q+j \ mod \ q}(t_{[\frac{i-1}{p}]+1,[\frac{j-1}{p}]+1})$$

This method is very easy to be implemented in a programming language using bitwise operations. Also, the memory space required to retain the matrix using this algorithm is

$$N + ([\frac{n-1}{p}] + 1) \times ([\frac{m-1}{q}] + 1).$$

This storage method is obviously better than classical methods if the numbers p and q are big enough.

Denoting

- r, the number which represents the memorising index of the given matrix A, ($r = \frac{N}{nm}$, $N \neq nm$) and
- $r^{'}$, the memorising index of the matrix A through this method

then, the inequality

$$pq > \frac{nm}{(r-1)N}$$

is a condition to obtain a better memorising index.

3. Conclusions

A new method to memorizing a sparse matrix was developed in this paper. Blocks of matrices were binary converted and were taken into consideration when the elements were accessed. The solving method was compared with other algorithms and a condition of efficiency was formulated. Computational results were presented in order to authenticate the proposed algorithm.

References

[1] Y. Saad, *Iterative Methods for Linear Systems*, PWS Publishing, Boston, 1996.

[2] N. Andrei, C. Răsturnoiu, *Matrice rare şi aplicaţiile lor*, Bucureşti, 1983.

[3] Pissanetzky, Sergio, *Sparse Matrix Technology*, Academic Press Inc., London, 1984.

Vlad Monescu
Department of Computer Science
Transilvania University of Brasov
50 Iuliu Maniu
email:*v.monescu@info.unitbv.ro*