Consiglio Nazionale delle Ricerche

ISTITUTO DI ELABORAZIONE DELLA INFORMAZIONE

PISA

COMPLEXITY MEASURES FOR MATRIX MULTIPLICATION ALGORITHMS.

F. Romani

Nota Interna
B79-9

Aprile

COMPLEXITY MEASURES FOR MATRIX MULTIPLICATION ALGORITHMS

Francesco Romani (+)

SUMMARY A new class of algorithms for the computation of bilinear forms has been recently introduced [1,3]. These algorithms approximate the result with an arbitrarily small error and their use may reduce the multiplicative complexity of some problems. This is the case of n x n matrix multiplication for which an approximate algorithm of complexity $O(n^{\log_{11} 1000})$ has been found. A comparison between approximate and exact algorithms has to take into account the complexity-stability relations.

In this paper some complexity measures for matrix multiplication algorithms are discussed and the exact and approximate algorithms are evaluated. Multiplicative complexity is shown to remain a valid comparison test and the cost of approximation appears to be only a logarithmic factor.

(+) Istituto di Elaborazione della Informazione, via S.Maria 46, 56100 Pisa, ITALY

1. INTRODUCTION AND PRELIMINARIES

Consider the problem of computing the bilinear forms

$$\underline{x}^T A_h \underline{y} \qquad (h=1, \ldots p)$$

$\underline{x}$ n-vector, $\underline{y}$ m-vector, $A_h \equiv \{a_{ij}^{(h)}\}$ n x m matrices.

In [1,3] those algorithms yielding an exact solution to this problem were called EC-algorithms (Exactly Computing).

A bilinear EC algorithm is identified by three matrices $U\equiv\{u_{ir}\}$, $V\equiv\{v_{jr}\}$, $W\equiv\{w_{hr}\}$ $(i=1,\ldots n; j=1,\ldots m; h=1,\ldots p, r=1,\ldots t)$ satisfying the condition

$$\sum_{r=1}^{t} u_{ir} v_{jr} w_{hr} = a_{ij}^{(h)}.$$

The bilinear forms are computed by the formula

$$\underline{x}^T A_h \underline{y} = \sum_{r=1}^{t} w_{hr} \Big(\sum_{i=1}^{n} u_{ir} x_i\Big)\Big(\sum_{j=1}^{m} v_{jr} y_j\Big) \qquad (h=1,\ldots p)$$

and t is the number of non scalar multiplications.

APA-algorithms (Arbitrary Precision Approximating) were introduced to take advantage of the circumstance that the complexity may be reduced by allowing the result to be affected by an arbitrarily small error. A bilinear APA-algorithm is identified by three matrices $U(\varepsilon), V(\varepsilon), W(\varepsilon)$ satisfying the condition

$$\sum_{r=1}^{t'} u_{ir}(\varepsilon) v_{jr}(\varepsilon) w_{hr}(\varepsilon) = a_{ij}^{(h)} + e_{ij}^{(h)}(\varepsilon) \quad (h=1,\ldots,p)$$

where $E_h(\varepsilon) \equiv \{e_{ij}^{(h)}(\varepsilon)\}$ is a matrix of continuous functions of ε and $E_h(0)$ are null matrices. If the entries of U,V,W are powers of ε the $E_h(\varepsilon)$ are polynomials in ε .

REMARK: In [1] an APA-algorithm for n x n matrix multiplication with multiplicative complexity $O(n^{2.7799})$ is presented.

In [5] it is shown that the existence of APA-algorithms requiring t' multiplications implies the existence of EC-algorithms requiring $t'(1+d)$ multiplications; d is the degree of the polynomial corrections $E_h(\varepsilon)$.

Matrix multiplication (in the following MM) is a special case of the problem of computing bilinear forms. Moreover the three-way array $\mathcal{A} = \{a_{ij}^{(h)}\}$ associated to a $k^q \times k^q$ MM problem is the q-th tensorial power of the three-way array associated to the k x k problem [5]. This fact supports the well known technique to derive a general n x n algorithm from a k x k algorithm by recursive partitioning [7,9]. As a matter of fact the recursive application of the same algorithm is equivalent to use a bilinear algorithm identified by matrices $U^{(q)}, V^{(q)}, W^{(q)}$ which are the q-th tensorial powers of U,V,W.

In section 2 some complexity measures to evaluate general MM algorithms by taking into account numerical stability are discussed. This allows to compare the efficiency of EC-algorithms and APA-algorithms. An alternative technique to perform MM using APA-algorithms is also considered. Logarithms are to base 2 throughout this paper, unless otherwise indicated.

2. COMPLEXITY MEASURES FOR MATRIX MULTIPLICATION ALGORITHMS

Some complexity measures for general $n \times n$ MM algorithms are discussed in this section. All of these measures are defined as the order of infinity of functions of n. A few of such measures are well known from the literature, the remaining have been recently introduced or are natural generalizations of known measures.

Multiplicative complexity M(n) [8].

This is defined as the order of non scalar multiplications required to calculate the product.

Operation complexity OP(n) [8].

This is the order of arithmetic operations required to calculate the product. If the algorithm makes use of recursive partitioning, OP(n) can be proved to be of the same order of M(n).

Fixed Precision complexity FP(n).

Sometimes it is required to compare algorithms with different numerical stabilities. In this case it is useful to take into account the cost of calculating the result with a given accuracy. We define the Fixed Precision complexity as follows:

$$FP(n) = OP(n) \;\; m(b(n,s_0))$$

where m(x) is the complexity of integer multiplication with x digits and b(n,s) the number of digits required in the arithmetic operations to obtain a relative error $e \leq 2^{-s}$. In [9] it is shown that $m(x)=O(x \log x \log \log x)$. The relative error of a general MM algorithm using b digits in the arithmetic can be written in the form

$$e \leq 2^{-b/\omega(n)} + \psi(n)$$

where $\omega(n)$ and $\psi(n)$ are functions depending on the algorithm; hence $b(n,s) = \omega(n)(s+\psi(n))$ and

$$FP(n)=O(OP(n)\; m(\omega(n)(\psi(n)+1))).$$

REMARK: Fixed Precision complexity is the most natural measure to estimate the real bit operation complexity of a MM algorithm.

Asymptotical complexity AC(n) [2].

Another way to compare algorithms while taking into account the numerical stability consists in evaluating the order of bit operations needed to compute the product with infinite precision. In order to achieve this result the measures of two algorithms a and b must have the following property

$$\frac{AC_a(n)}{AC_b(n)} = \lim_{s\to\infty} \frac{OP_a(n)\; m(b_a(n,s))}{OP_b(n)\; m(b_b(n,s))} .$$

This can be obtained by defining

$$AC(n) = \lim_{s\to\infty} OP(n) \frac{m(b(n,s))}{m(s)} .$$

Under some regularity hypotheses for m(x) it is proved that

$$AC(n) = OP(n)\,\omega(n), \; [2].$$

3. COMPLEXITY OF EC-ALGORITHMS CONTRUCTED BY RECURSIVE PARTITIONING

Consider an EC-algorithm for k x k MM requiring t multiplications and let $\alpha=\log_k t$. It is well known that a general algorithm can be contructed by applying the technique of recursive partitioning [8,10] and

$$OP(n) = O(M(n)) = O(n^{\alpha}).$$

On the other hand the relative error can be proved to be $O(n^{\gamma} 2^{-b})$ i.e. $\omega(n)=1$, $\psi(n)=\gamma \log n$. This yields

$$FP(n) = O(n^{\alpha} m(\log n)),$$

$$AC(n) = O(n^{\alpha}).$$

4. COMPLEXITY OF APA-ALGORITHMS

Consider an Arbitrary Precision Approximating algorithm for k x k MM with multiplicative complexity t; let $\beta=\log_k t$. The accuracy of the result depends on the number of digits used in the arithmetic and on the choice of ε .

There are two sources of errors when using APA-algorithms namely the error produced by floating point arithmetic and the error due to the algorithm itself. Assuming that $U(\varepsilon)$, $V(\varepsilon)$,$W(\varepsilon)$ are matrices of powers of ε let ε^{φ} be the highest infinite in $U(\varepsilon)$,$V(\varepsilon)$,$W(\varepsilon)$ and ε^{σ} the lowest infinitesimal in the correction matrices $E_h(\varepsilon)$ when $\varepsilon \to 0$. The order of the error due to arithmetic is $O(\varepsilon^{\varphi} 2^{-b})$ and the error of the algorithm is $O(\varepsilon^{\sigma})$. The best choice for ε is $2^{-\xi}$, $\xi=b/(\sigma-\varphi)$ and the overall error becomes $O(2^{-b\sigma/(\sigma-\varphi)})$ [4].

In order to obtain a general n x n algorithm the APA-algorithm is used recursively $\lceil \log_k n \rceil$ times, thus obtaining an operation complexity $OP(n)=O(n^{\beta})$.

In the general algorithm the highest infinite order in the resulting matrices U,V,W becomes $\varepsilon^{\varphi \log_k n}$ and the lowest infinitesimal in the correction remains ε^{σ} . Then the overall error is

$$O(n^{\gamma} 2^{-b\delta/(\delta-\varphi\log_k n)}) = O(2^{-b/\omega(n) + \psi(n)}) \quad \text{with } \omega(n)=O(\log n), \; \psi(n)=O(\log n),$$

the term n^{γ} takes into account the growth of errors not depending on ε. Hence the Fixed Precision complexity of APA-algorithms becomes

$$FP(n) = O(n^{\beta} m(\log^2 n)),$$

and the Asymptotic complexity

$$AC(n) = O(n^{\beta} \log n) \quad [1].$$

5. COMPLEXITY OF EC-ALGORITHMS DERIVED FROM APA-ALGORITHMS

As shown in [5] it is possible to derive EC-algorithms from APA-algorithms. Namely if the entries of the correction matrices $E_h(\varepsilon)$ are polynomials in ε with highest degree d, the resulting ECD-algorithm (Exactly Computing Derived) has a multiplicative complexity (1+d)t, where t is the multiplicative complexity of the APA-algorithm.

Consider an APA-algorithm $\mathfrak{a}$ for k x k MM requiring t multiplications; if $\mathfrak{a}$ is applied recursively $\log_k n_o$ times a general APA-algorithm results, and the entries of the correction matrices have highest degree $d \log_k n_o$. This implies that an ECD-algorithm with multiplicative complexity

$$M(n_o) = n_o^{\log_k t} (1+d \log_k n_o)$$

can be derived.

Then there exists a sequence of general algorithms with complexities

$$OP(n) = O(n^{\beta + \delta(n_o)})$$
$$FP(n) = O(n^{\beta + \delta(n_o)} m(\log n))$$
$$AC(n) = O(n^{\beta + \delta(n_o)})$$

where $\delta(n_o) = \log(1+d \log_k n_o) / \log n_o$ may be arbitrarily small

Another approach to construction of general EC-algorithms produces a better result; this approach is based on the same technique of [5]. Let $[\alpha_1, \alpha_2, \dots \alpha_{\tilde{d}+1}]$ be the solution of the Vandermonde system:

$$\begin{bmatrix} 1 & 1 & \dots & 1 \\ \varepsilon_1 & \varepsilon_2 & \dots & \varepsilon_{\tilde{d}+1} \\ \vdots & & & \vdots \\ \varepsilon_1^{\tilde{d}} & \varepsilon_2^{\tilde{d}} & \dots & \varepsilon_{\tilde{d}+1}^{\tilde{d}} \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_{\tilde{d}+1} \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

where $\tilde{d} = d \log_k n$. Since the entries of the correction matrices are polynomials in ε of degree $\leq \tilde{d}$ and the terms of degree 0 are equal to 0, we have

$$\sum_{i=1}^{\tilde{d}+1} \alpha_i = 1, \qquad \sum_{i=1}^{\tilde{d}+1} \alpha_i E_h(\varepsilon_i) = 0 \qquad (h=1,2,\dots n^2),$$

and

$$\sum_{i=1}^{\tilde{d}+1} \alpha_i \underline{x}^T(A_h + E_h(\varepsilon_i))\underline{y} = \underline{x}^T A_h \underline{y} \qquad (h=1,2,\dots n^2).$$

The n x n APA-algorithm has to be executed $d \log_k n + 1$ times with different values of ε and the results have to be linearly combined. This kind of algorithm is called ECC (Exactly Computing Corrected).

The multiplicative complexity is then $O(n^\beta \log n)$.

The best choice for the values of ε is $\varepsilon_i = i$ $(i=1,2,\dots d+1)$ since it minimizes the number of digits needed to represent the ε_i's and the corresponding values of α are easily computable; namely

$$\alpha_i = (-1)^{i+1} \binom{\tilde{d}+1}{i}, \quad [6].$$

Using ECC-algorithms the error due to the approximation disappears and the error due to floating point arithmetic has the form

$$e = O(\varepsilon^{\varphi \log_K n}\, 2^{-b}).$$

From $\varepsilon = O(\log n)$ it follows

$$e = O(2^{-b + \varphi \log_K n \log \log n}) \quad \text{i.e.} \quad \omega(n) = O(1),$$

$$\psi(n) = O(\log n \log \log n).$$

The Fixed Precision complexity becomes

$$FP(n) = O(n^{\beta} \log n\; m(\log n \log \log n))$$

and the Asymptotic complexity

$$AC(n) = O(n^{\beta} \log n)$$

6. CONCLUSION

Table I displays a comparison of the complexity measures for the different types of algorithms.
Note that the main parameter to evaluate Matrix Multiplication algorithms remains the number of non scalar multiplications. The overhead due to numerical instability using approximate algorithms consists of logarithmic factors. Moreover APA-algorithms appear to be less complex than the corresponding ECD and ECC algorithms.

TABLE I

	OP	FP	AC	
EC	$O(n^{\alpha})$	$O(n^{\alpha} m(\log n))$	$O(n^{\alpha})$	
APA	$O(n^{\beta})$	$O(n^{\beta} m(\log^2 n))$	$O(n^{\beta} \log n)$	
ECD	$O(n^{\beta+\varepsilon})$	$O(n^{\beta+\varepsilon} m(\log n))$	$O(n^{\beta+\varepsilon})$	$\forall \varepsilon > 0$
ECC	$O(n^{\beta} \log n)$	$O(n^{\beta} \log n \; m(\log n \log \log n))$	$O(n^{\beta} \log n)$	

Table I. Complexities of MM algorithms. The best known value for α is 2.7951[7] and for β is $\log_{12} 1000 = 2.7799..$[1].

REFERENCES

1. D.BINI, M.CAPOVANI, G.LOTTI, F.ROMANI, $O(n^{2.7799})$ Complexity for n x n Matrix Multiplication. IEI Report B78-27, (1978).
2. D.BINI, G.LOTTI, F.ROMANI, Stability and Complexity in the Evaluation of a Set of Bilinear Forms. IEI Report B78-25, (1978).
3. D.BINI, G.LOTTI, F.ROMANI, Suboptimal Solutions for the Bilinear Form Computational Problem. IEI Report B78-26, (1978).
4. D.BINI, Border Tensorial Rank of Triangular Toeplitz Matrices. IEI Report B78-28, (1978).
5. D.BINI, Relations between Exact and Approximate Bilinear Algorithms. Applications. (1979). (Submitted to Calcolo).
6. R.GREGORY, D.KARNEY, A Collection of Matrices for Testing Computational Algorithms. (1969), Interscience, New York.
7. V.PAN, Strassen's Algorithm is not Optimal, Trilinear Technique of Aggregating, Uniting and Canceling for Constructing Fast Algorithms for Matrix Operations. IEEE 19-th Annual Symposium on Foundations of Computer Science, (1978), 166-176.
8. M.S.PATERSON, Complexity of Matrix Algorithms, in: J.W. DE BAKKER ed., Foundations of Computer Science, Mathematical Centre Tracts, 63 (1975), 181-215.
9. A.SCHÖNHAGE, V.STRASSEN, Schnelle Multiplication Grosser Zahlen. Computing, 7 (1971) 281-292.
10. V.STRASSEN, Gaussian Elimination is not Optimal. Numer. Math., 13 (1969) 354-356.