Advanced Algorithms	**Lecture 13 : May 10, 1995**
Lecturer: Johan Håstad	*Scribe: Christer Berg*

Matrix multiplication

13.1 Introduction

Given two $n \times n$ matrices, A and B, we wish to examine how fast we can compute the product $AB = C$. The normal method takes time $O(n^3)$ since each entry in C is computed as

$$c_{i,k} = \sum_{j=1}^{n} a_{i,j} b_{j,k},$$

which means that each of the n^2 entries in C takes time $O(n)$ to compute.

The first algorithm to do better than this obvious bound was given by Strassen in 1969 [3]. The idea of this algorithm is to divide each $n \times n$ matrix into four parts of size $n/2 \times n/2$. If we view the submatrices of size $n/2 \times n/2$ as single elements, the problem is reduced to computing the product of two 2×2 matrices, and each subproblem can be solved recursively. The fundamental part of the algorithm is thereby the computation of the product of two 2×2 matrices and the number of multiplications this requires (the addition of matrices is much less costly than the multiplication). While the ordinary method to multiply two 2×2 matrices uses 8 multiplication, Strassen found a way that uses only seven multiplications together with 18 additions.

If $T(n)$ denotes the time this algorithm takes to multiply two $n \times n$ matrices, we hence get the recurrence relation $T(n) = 7T(n/2) + 18n^2$ which is solved by $T(n) \in O(n^{\log_2 7}) \in O(n^{2.81})$.

Today, the fastest known algorithm is by Coppersmith and Winograd [1] (1987), and runs in time $O(n^{2.376})$. In these notes we will describe a algorithm with complexity $O(n^{2.67})$. Note, however, that these two algorithms are not practical.

13.2 Bilinear problems

A problem on the following form is called *bilinear*. Input: x_i, $i \in I$, and y_j, $j \in J$ where I and J are index sets. Output: z_k, $k \in K$, where K is an index set and

$$z_k = \sum_{i,j} c_{i,j,k} x_i y_j.$$

Example 1 *The multiplication of two complex numbers $x_1 + ix_2$ and $y_1 + iy_2$ can be written as $(y_1 + ix_2)(y_1 + iy_2) = z_1 + iz_2$ where*

$$\begin{aligned} z_1 &= x_1 y_1 - x_2 y_2, \\ z_2 &= x_1 y_2 + x_2 y_1. \end{aligned}$$

Comparing this with the sum z_k above, we see that the coefficients $c_{i,j,1}$ can be written as the matrix

$$\begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix},$$

and $c_{i,j,2}$ as

$$\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}.$$

Hence these two matrices defines the output in terms of a three dimensional array — this array will be called a tensor.

Multiplication of two complex numbers performed as above requires four multiplications of real numbers — is it possible to do this with only three multiplications? Let

$$\begin{aligned} p_0 &= x_1 y_1, \\ p_1 &= x_2 y_2, \\ p_2 &= (x_1 + x_2)(y_1 + y_2), \end{aligned}$$

then $z_1 = p_0 - p_1$ and $z_2 = p_2 - p_0 - p_1$ which means that we, in fact, can do complex multiplication by using only three ordinary multiplications. However, it can be shown (and this is one of the homework problems) that two multiplications will not suffice.

We will see that these facts will correspond to the *rank* of a tensor. To motivate this concept, consider the contribution of p_0 to the output, namely

$$\begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} -1 & 0 \\ 0 & 0 \end{pmatrix},$$

the contribution of p_1 is

$$\begin{pmatrix} 0 & 0 \\ 0 & -1 \end{pmatrix} \begin{pmatrix} 0 & 0 \\ 0 & -1 \end{pmatrix},$$

and for p_2, the contribution is

$$\begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix}.$$

Note now that the addition of these tensors gives the tensor of our problem, that is

$$\begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}.$$

Definition 1 *A tensor has* rank 1 *if $c_{i,j,k} = d_i e_j f_k$ for some vectors d, e and f.*

This definition carries over to ordinary matrices, since there we have that $m_{i,j} = a_i b_j$ for vectors a and b if the rank is one. This follows since if the rank is 1 then each row is proportional to exactly one of the rows and so we can choose a as this row and in vector b we just fill in the proportionality constants.

Definition 2 *The* rank *of a tensor, C, is the least number of rank 1 tensors, N_i, such that $C = \sum_i N_i$.*

Also here the same definition holds for ordinary matrices.

Theorem 1 *The rank of a tensor equals the number of multiplications that is required to compute the underlying computational problem.*

Proof: First, note that a tensor of rank 1 in a natural way corresponds to a computation consisting of 1 multiplication. Next, there is a one-to-one correspondence between the decomposition of the tensor into rank 1 tensors and the corresponding multiplications in the computational problem. Together this proves the theorem. ■

Hence, this definition implies that it would be useful to be able to compute the rank of a tensor, but while computing the rank of a matrix can be achieved quickly by Gaussian elimination, for a tensor the problem is NP-complete [2].

Note that the rank of a tensor depends on the field over which it is computed. In particular, the rank is in general different over $\mathbb{R}$ than over $\mathbb{C}$.

Example 2 *Consider the tensor*

$$\begin{pmatrix} 1 & i \\ i & -1 \end{pmatrix} \begin{pmatrix} -i & 1 \\ 1 & i \end{pmatrix},$$

and

$$\begin{pmatrix} 1 & -i \\ -i & -1 \end{pmatrix} \begin{pmatrix} i & 1 \\ 1 & -i \end{pmatrix},$$

which both have rank 1, since in the first case we can choose $d = (1, i)$, $e = (1, i)$, *and* $f = (1, -i)$ *and in the second case we have* $d = (1, -i)$, $e = (1, -i)$, *and* $f = (1, i)$. *Adding these together gives the tensor for computing complex multiplication. Hence this tensor has rank 3 over* $\mathbb{R}$ *but rank 2 over* $\mathbb{C}$.

An alternative, but equivalent, way to view a tensor is as a collection of matrices, M_i, and the rank is then the least number of rank 1 matrices, N_j, such that each M_i can be written as a linear combination of the N_j's.

Example 3 *The tensor*

$$\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix},$$

corresponds to

$$\begin{aligned} z_1 &= x_1 y_1 + x_2 y_2, \\ z_2 &= x_1 y_2. \end{aligned}$$

We see that the computation of z_1 *and* z_2 *can be performed with 3 multiplications — what about 2 multiplications? If this is possible the rank of the tensor must be 2, which means that the two matrices of the tensor can be written as linear combinations of two rank 1 matrices* N_1 *and* N_2. *We may suppose that* $N_1 = \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix}$. *In this case we must have*

$$\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} = a \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix} + N_2 \;\Rightarrow\; N_2 = \begin{pmatrix} 1 & -a \\ 0 & 1 \end{pmatrix}.$$

However, for all values of a, *the rank of* N_2 *is greater than 1, which implies that the given tensor can not be written as linear combination of only two rank 1 matrices. Hence, the rank of the given tensor is 3, and the computation of* z_1 *and* z_2 *then must take 3 multiplications.*

Example 4 *Consider again the tensor given in the previous example. For any ε the tensor*

$$\begin{pmatrix} 0 & \varepsilon^{-1} \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 0 & 1 \\ 0 & \varepsilon \end{pmatrix},$$

and

$$\begin{pmatrix} 1 & -\varepsilon^{-1} \\ 0 & 0 \end{pmatrix} \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix},$$

has rank 1. Adding these together yields

$$\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 0 & 1 \\ 0 & \varepsilon \end{pmatrix}.$$

So by making ε small, we may produce a tensor arbitrarily close to the tensor of the previous example. This calls for the following definition.

Definition 3 *The* limit rank *of a tensor, C, is equal to t if it arbitrarily close to C exists tensors of rank t.*

From Example 2 and 4 we thus have a tensor whose rank is 3 but whose limit rank is 2. For ordinary matrices however, the limit rank is always equal to the rank.

The maximal rank of an $n \times n$ matrix is of course n, but the exact value of the maximal rank, r, of an $n \times n \times n$ tensor is unknown. The value of r is known only up to the inequality $n^2/3 \le r \le n^2/2$.

13.3 Matrix multiplication and tensors

To return to the main subject of this lecture, we want to compute

$$\begin{pmatrix} x_{1,1} & x_{1,2} \\ x_{2,1} & x_{2,2} \end{pmatrix} \begin{pmatrix} y_{1,1} & y_{1,2} \\ y_{2,1} & y_{2,2} \end{pmatrix} = \begin{pmatrix} z_{1,1} & z_{1,2} \\ z_{2,1} & z_{2,2} \end{pmatrix},$$

and now we may ask what tensor this corresponds to. We have, for example, that $z_{1,1} = x_{1,1}y_{1,1} + x_{1,2}y_{2,1}$, which can be expressed as a 4×4 matrix with rows labeled $x_{1,1}$, $x_{1,2}$, $x_{2,1}$, $x_{2,2}$, and columns labeled $y_{1,1}$, $y_{1,2}$, $y_{2,1}$, $y_{2,2}$, that is,

$$z_{1,1} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}.$$

Hence, we get the tensor

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix},$$

where the matrices correspond to $z_{1,1}$, $z_{1,2}$, $z_{2,1}$ and $z_{2,2}$, respectively.

In general we have that each entry in the matrix product is

$$z_{i,k} = \sum_{j=1}^{n} x_{i,j} y_{j,k}.$$

Note that this is notation actually is a simplification, since we really have n^2 variables each of z_{k_1,k_2}, x_{i_1,i_2}, and y_{j_1,j_2}, and hence we should write the bilinear problem as

$$z_{k_1,k_2} = \sum_{i_1,i_2,j_1,j_2,k_1,k_2} c_{(i_1,i_2),(j_1,j_2),(k_1,k_2)} x_{i_1,i_2} y_{j_1,j_2},$$

where $c_{(i_1,i_2),(j_1,j_2),(k_1,k_2)} = 1$ if $i_1 = k_1$, $j_1 = i_2$, and $k_2 = j_2$, and 0 otherwise. For example, for $z_{1,1}$ we get $c_{(1,i_2),(j_1,1),(1,1)} = 1$ whenever $i_2 = j_1$, as we expect.

We will find it convenient to express this as a formal sum,

$$\sum_{i,j,k} x_{i,j} y_{j,k} v_{i,k},$$

that is, instead of having the variables $z_{i,k}$ as output we introduce formal variables $v_{i,k}$ and move them into the sum. (As an analogy, consider a vector $(a_0, a_1, \ldots, a_{n-1})$ which can be written as a formal power series $a_0 + a_1 t + \cdots + a_{n-1} t^{n-1}$.) Note that below the formal variables are called $z_{i,k}$.

Definition 4 *Let $T(m,p,r)$ denote the tensor that corresponds to the multiplication of an $m \times p$ matrix with a $p \times r$ matrix, that is,*

$$T(m,p,r) = \sum_{i=1}^{m} \sum_{j=1}^{p} \sum_{k=1}^{r} x_{i,j} y_{j,k} z_{i,k},$$

and let $M(m,p,r)$ denote the number of multiplications that this matrix multiplication requires, that is, $M(m,p,r)$ is the rank of $T(m,p,r)$ (For our original problem, we are hence interested in knowing the value of $M(n,n,n)$.)

Theorem 2 *We have that $M(m,p,r) = M(p,r,m)$, and likewise for all permutations of m, p, and r.*

Proof: Since

$$T(m,p,r) = \sum_{i=1}^{m}\sum_{j=1}^{p}\sum_{k=1}^{r} x_{i,j}y_{j,k}z_{i,k},$$

and

$$T(p,r,m) = \sum_{j=1}^{p}\sum_{k=1}^{r}\sum_{i=1}^{m} x_{j,k}y_{k,i}z_{j,i},$$

we see that $T(m,p,r)$ and $T(p,r,m)$ are equal up to a permutation of the indices and this permutation does not change the rank. ■

Definition 5 *Given tensors T_1 and T_2 which can be written as $c^1_{i_1,j_1,k_1}$ and $c^2_{i_2,j_2,k_2}$, respectively, where $i_1 \in I_1$, $j_1 \in J_1$, $k_1 \in K_1$, $i_2 \in I_2$, $j_2 \in J_2$, and $k_2 \in K_2$. The* tensor product*, $T_1 \otimes T_2$, is a tensor with index sets $(i_1,i_2) \in I_1 \times I_2$, $(j_1,j_2) \in J_1 \times J_2$, and $(k_1,k_2) \in K_1 \times K_2$, such that $c_{(i_1,i_2),(j_1,j_2),(k_1,k_2)} = c^1_{i_1,j_1,k_1} \cdot c^2_{i_2,j_2,k_2}$.*

Theorem 3 *For two tensors, T_1 and T_2, we have that*

$$\mathrm{rank}(T_1 \otimes T_2) \leq \mathrm{rank}(T_1)\mathrm{rank}(T_2).$$

Proof: Let $r_1 = \mathrm{rank}(T_1)$, that is, we have

$$T_1 = \sum_{i=1}^{r_1} N^1_i,$$

where each N^1_i is a tensor of rank 1. Correspondingly, for T_2 we have $T_2 = \sum_{j=1}^{r_2} N^2_j$. This means that

$$T_1 \otimes T_2 = \sum_{i=1}^{r_1} N^1_i \otimes \sum_{j=1}^{r_2} N^2_j = \sum_{i=1}^{r_1}\sum_{j=1}^{r_2} N^1_i \otimes N^2_j,$$

since $\otimes$ is distributive over $+$. It is easy to check that $N^1_i \otimes N^2_j$ is a tensor of rank 1, and hence, the theorem follows. ■

Theorem 4 *For two tensors, $T(m_1,p_1,r_1)$ and $T(m_2,p_2,r_2)$, we have that*

$$T(m_1,p_1,r_1) \otimes T(m_2,p_2,r_2) = T(m_1m_2, p_1p_2, r_1r_2).$$

Proof: Let

$$T(m_1, p_1, r_1) = \sum_{i_1=1}^{m_1} \sum_{j_1=1}^{p_1} \sum_{k_1=1}^{r_1} x_{i_1,j_1} y_{j_1,k_1} z_{i_1,k_1},$$

and

$$T(m_2, p_2, r_2) = \sum_{i_2=1}^{m_2} \sum_{j_2=1}^{p_2} \sum_{k_2=1}^{r_2} x_{i_2,j_2} y_{j_2,k_2} z_{i_2,k_2}.$$

Then

$$\begin{aligned} T(m_1, p_1, r_1) \otimes T(m_2, p_2, r_2) &= \sum_{i_1,i_2} \sum_{j_1,j_2} \sum_{k_1,k_2} x_{(i_1 i_2, j_1 j_2)} y_{(j_1 j_2, k_1 k_2)} z_{(i_1 i_2, k_1 k_2)} \\ &= T(m_1 m_2, p_1 p_2, r_1 r_2), \end{aligned}$$

which is what we wanted to show. ■

Using the notation we have introduced, and the two theorems above, we can state Strassen's result as follows. Suppose n is even. Then Theorem 4 implies that $T(n, n, n) = T(n/2, n/2, n/2) \otimes T(2, 2, 2)$ and Theorem 3 says that $M(n, n, n) \leq M(n/2, n/2, n/2)M(2, 2, 2) = M(n/2, n/2, n/2) \cdot 7$.

References

[1] D. Coppersmith and S. Winograd. Matrix multiplication via arithmetic progressions. *19th Annual ACM Symposium on Theory of Computing*, 1–6, 1987.

[2] J. Håstad. Tensor Rank is NP-Complete. *J. of Algorithms* **11**, 644–654, 1990.

[3] V. Strassen. Gaussian elimination is not optimal. *Numerische Mathematik*, 13, 1969.