Complexity of Matrix Multiplication and Tensor Rank

KIAS, Seoul

February 27, 2014

Giorgio Ottaviani

Università di Firenze

Plan of the talk

- Is the way we multiply two one variables polynomials computationally optimal ?
  $(f(x), g(x)) \mapsto f(x) \cdot g(x)$

- Is the way we multiply two matrices computationally optimal ?
  $(A, B) \mapsto A \cdot B$

- Some results obtained with tools from representation theory and algebraic geometry.

Multiplying two linear polynomials

We multiply $a_0 + a_1 x$ by $b_0 + b_1 x$, where $a_i, b_j \in \mathbb{C}$. Get

$$\underbrace{a_0 b_0}_{1} + (\underbrace{a_0 b_1}_{2} + \underbrace{a_1 b_0}_{3})x + \underbrace{a_1 b_1}_{4} x^2$$

4 complex multiplications and 3 additions are needed.

Only 3 complex multiplications.

May we multiply $a_0 + a_1 x$ by $b_0 + b_1 x$ with just 3 complex multiplications ? We may proceed in the following way (Karatsuba, 1960).

$$(a_0 + a_1 x)(b_0 + b_1 x) =$$

$$\underbrace{a_0 b_0}_{1} + \left(-\underbrace{a_0 b_0}_{1} + \underbrace{(a_0 + a_1)(b_0 + b_1)}_{2} - \underbrace{a_1 b_1}_{3} \right) x + \underbrace{a_1 b_1}_{3} x^2$$

so we conclude with just 3 complex multiplications, but with 6 additions.

Iterating is surprising.

Standard algorithm requires 4 multiplications and 3 additions.
Karatsuba algorithm requires 3 multiplications and 6 additions.

Is it convenient ?

Karatsuba discovery: iterating the algorithm, it becomes convenient!

Karatsuba algorithm (1960)

Multiply, for simplicity, polynomials of degree $n = 2^m - 1$.
Write $a(x) = a_0(x) + a_1(x)x^{(n+1)/2}$, with $\deg a_i(x) \leq 2^{m-1} - 1$,
$b(x) = b_0(x) + b_1(x)x^{(n+1)/2}$, with $\deg b_i(x) \leq 2^{m-1} - 1$.
Need 3 multiplications and 6 additions of polynomials of half degree.
Indeed we get,

$$(a_0(x) + a_1(x)x^{(n+1)/2})(b_0(x) + b_1(x)x^{(n+1)/2}) = \underbrace{a_0(x)b_0(x)}_{1} +$$

$$\left(-\underbrace{a_0(x)b_0(x)}_{1} + \underbrace{(a_0(x)+a_1(x))(b_0(x)+b_1(x))}_{2} - \underbrace{a_1(x)b_1(x)}_{3}\right) x^{(n+1)/2} +$$

$$\underbrace{a_1(x)b_1(x)}_{3} x^{n+1}$$

Counting arithmetic operations

Iterating, in degree $2^m - 1$, we have 3^m complex multiplications and $6 \cdot 2^{m-1}$ complex additions.

$$\text{Total cost of arithmetic operations} = \underbrace{3^m}_{\text{leading term}} + 6 \cdot 2^{m-1}$$

The number of additions does not contribute asymptotically !
So, in degree $n \gg 0$, we get the total cost is $O(n^{\log_2 3}) = O(n^{1.585})$, better than $O(n^2)$ of the naive algorithm. Karatsuba answered a problem posed by Kolmogorov and invented the method called

"*Divide and conquer*"

Understanding Karatsuba algorithm, the Discrete Fourier Transform

There is an analogous, cheap way to multiply two polynomials $a(x)$ of degree α and $b(x)$ of degree β, dating back to Gauss.
Evaluate $a(x)$ for $x_1, \ldots, x_{\alpha+\beta+1}$ and get the vector

$$(a(x_1), \ldots, a(x_{\alpha+\beta+1})).$$

Evaluate $b(x)$ and get

$$(b(x_1), \ldots, b(x_{\alpha+\beta+1})).$$

Multiply pointwise the two vectors

$$(a(x_1)b(x_1), \ldots, a(x_{\alpha+\beta+1})b(x_{\alpha+\beta+1}))$$

and interpolate, to get the wished polynomial of degree $\alpha + \beta$.
When the points are $(\alpha + \beta + 1)$-th roots of unity, this is essentially the *Discrete Fourier Transform.*

Karatsuba algorithm revisited

Come back to multiplication of $a(x) = a_0 + a_1 x$ by $b(x) = b_0 + b_1 x$.
We proceed in the following way. Evaluate $a(x)$ for $x = 0, 1, \infty$ and get the vector

$$(a_0, a_0 + a_1, a_2).$$

Evaluate b and get

$$(b_0, b_0 + b_1, b_2).$$

Multiplying pointwise, get $a(x)b(x) = c(x)$
$c(0) = a_0 b_0$,
$c(1) = (a_0 + a_1)(b_0 + b_1)$,
$c(\infty) = a_2 b_2$. To go back to c_0, c_1, c_2 we need to interpolate (invert the evaluation) and we get exactly Karatsuba formulas.

Geometrical interpretation

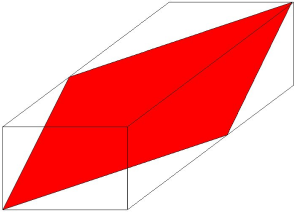
Polynomial multiplication is a bilinear map

$$\begin{array}{rcl} S^{\alpha}\mathbb{C}^2 \times S^{\beta}\mathbb{C}^2 & \to & S^{\alpha+\beta}\mathbb{C}^2 \\ (a(x), b(x)) & \mapsto & a(x)b(x) \end{array}$$

encoded in a tensor

$$t \in S^{\alpha}\mathbb{C}^{2^\vee} \otimes S^{\beta}\mathbb{C}^{2^\vee} \otimes S^{\alpha+\beta}\mathbb{C}^2 = A \otimes B \otimes C,$$

where S^p denotes p-th symmetric power.
The tensor t is familiar, its 3dimensional shape is diagonal



Ubiquity of polynomial multiplication tensor

The tensor t can be represented as the Hankel matrix

$$\begin{pmatrix} c_0 & c_1 & \dots & c_\alpha \\ c_1 & & \iddots & c_{\alpha+1} \\ \vdots & \iddots & \iddots & \vdots \\ & & & \vdots \\ c_\alpha & & & \\ \vdots & & & \vdots \\ c_\beta & \dots & \dots & c_{\alpha+\beta} \end{pmatrix}$$

In case $\alpha = \beta = 1$ the tensor t has 4 decomposable summands (with obvious notations)

$$t = \underbrace{a_0 b_0 c_0}_{1} + \underbrace{a_1 b_0 c_1}_{2} + \underbrace{a_0 b_1 c_1}_{3} + \underbrace{a_1 b_1 c_2}_{4}$$

Rank of a tensor := minimum number of decomposable tensors needed to express it as a sum.

For matrices (2-ways tensors) it coincides with the usual rank.

So $\mathrm{rk}(t) \leq 4$.

The Karatsuba decomposition

Karatsuba algorithm corresponds to the following tensor decomposition

$$t = \underbrace{a_0 b_0 c_0}_{1} + \underbrace{a_1 b_0 c_1}_{2} + \underbrace{a_0 b_1 c_1}_{3} + \underbrace{a_1 b_1 c_2}_{4} =$$

$$\underbrace{(c_0 - c_1) a_0 b_0}_{1} + \underbrace{c_1 (a_0 + a_1)(b_0 + b_1)}_{2} + \underbrace{(-c_1 + c_2) a_1 b_1}_{3}$$

which is called the Karatsuba decomposition.

This explains that the tensor t has rank 3.

Rank of a tensor = fair measure of its complexity.

In the tensor space $A \otimes B \otimes C$ there is the Segre variety of decomposable (rank 1) tensors $X = \mathbb{P}(A) \times \mathbb{P}(B) \times \mathbb{P}(C)$.
Tensors of rank 2, like $a_0b_0c_0 + a_1b_1c_1$, lie in the line joining $a_0b_0c_0$ and $a_1b_1c_1$.
Tensors of rank k lie in the span of k points on the Segre variety.

The k-secant variety $\sigma_k(X)$ is the Zariski closure

$$\sigma_k(X) := \overline{\bigcup_{x_1,\ldots x_k \in X} \langle x_1, \ldots, x_k \rangle}$$

it is the smallest algebraic variety containing all tensors of rank k.

Decomposition according to DFT in cubic case

By choosing $\tau = e^{\frac{2\pi\sqrt{-1}}{3}}$
and the three points as the three roots of unity $1, \tau, \tau^2$
we get the more symmetric decomposition

$$t = \underbrace{a_0 b_0 c_0}_{1} + \underbrace{a_1 b_0 c_1}_{2} + \underbrace{a_0 b_1 c_1}_{3} + \underbrace{a_1 b_1 c_2}_{4} =$$

$$\frac{1}{3}[\underbrace{(c_0 + c_1 + c_2)(a_0 + a_1)(b_0 + b_1)}_{1} +$$

$$\underbrace{(c_0 + c_1\tau^2 + c_2\tau)(a_0 + a_1\tau)(b_0 + b_1\tau)}_{2} +$$

$$\underbrace{(c_0 + c_1\tau + c_2\tau^2)(a_0 + a_1\tau^2)(b_0 + b_1\tau^2)]}_{3}$$

The discrete Fourier transform

Indeed, in general, we have the following description.
Let $z_0, \dots z_{\alpha+\beta}$ be distinct values. Let $y_i = \sum_{j=0}^{\alpha+\beta} c_j (z_i)^j$, for $i = 0, \dots, \alpha+\beta$, be inverted by $c_i = \sum_{j=0}^{\alpha+\beta} y_j v_{ij}$, then the multiplication tensor is

$$\sum_{j=0}^{\alpha+\beta} \left[\underbrace{\left(\sum_{i=0}^{\alpha+\beta} c_i v_{ji}\right)\left(\sum_{i=0}^{\alpha} a_i z_j^i\right)\left(\sum_{i=0}^{\beta} b_i z_j^i\right)}_{j\text{-th summand}} \right] = \sum_{i=0}^{\alpha} \sum_{j=0}^{\beta} a_i b_j c_{i+j} \qquad (1)$$

In the DFT, the points z_i are the $(\alpha+\beta+1)$-th roots of unity. In this case we have the Vandermonde matrix with (i,j) entry $e^{2\pi\sqrt{-1}\frac{ij}{\alpha+\beta+1}}$ which has the inverse with (i,j) entry $\frac{1}{\alpha+\beta+1} e^{2\pi\sqrt{-1}\frac{-ij}{\alpha+\beta+1}}$,

The discrete Fourier transform, II

so that we have the remarkable decomposition, $\tau = e^{\frac{2\pi\sqrt{-1}}{\alpha+\beta+1}}$

$$\frac{1}{\alpha+\beta+1}\sum_{j=0}^{\alpha+\beta}\left[\underbrace{(\sum_{i=0}^{\alpha+\beta} c_i\tau^{-ij})(\sum_{i=0}^{\alpha} a_i\tau^{ij})(\sum_{i=0}^{\beta} b_i\tau^{ij})}_{j\text{-th summand}}\right] = \sum_{i=0}^{\alpha}\sum_{j=0}^{\beta} a_i b_j c_{i+j} \tag{2}$$

Computation of polynomial multiplication is efficient due to Fast Fourier Transform (Cooley - Tukey, 1965) which divides iteratively even and odd terms in above sums.

Tensor rank of polynomial multiplication

In conclusion we have

Theorem (Fiduccia, Zalcstein)

- *The multiplication tensor in* $S^{\alpha}\mathbb{C}^2 \otimes S^{\beta}\mathbb{C}^2 \otimes S^{\alpha+\beta}\mathbb{C}^2$ *has rank* $\alpha + \beta + 1$.
- *The multiplication tensor in* $S^{n}\mathbb{C}^2 \otimes S^{n}\mathbb{C}^2 \otimes S^{2n}\mathbb{C}^2$ *of polynomials of degree* n *has rank* $2n + 1$.

The rank measures, asymptotically, the complexity of the polynomial multiplication. Note in above theorem the rank asymptotically corresponds to the size of the output of the algorithm, hence it cannot be improved.

Relevance of matrix multiplication algorithm

Many numerical algorithms use matrix multiplication. The complexity of matrix multiplication algorithm is crucial in many numerical routines.

$$M_{m,n} = \text{space of } m \times n \text{ matrices}$$

Matrix multiplication is a bilinear map

$$\begin{array}{ccc} M_{m,n} \times M_{n,l} & \rightarrow & M_{m,l} \\ (A, B) & \mapsto & A \cdot B \end{array}$$

where $A \cdot B = C$ is defined by $c_{ij} = \sum_k a_{ik} b_{kj}$.
This usual way to multiply a $m \times n$ matrix with a $n \times l$ matrix requires mnl multiplications and $ml(n-1)$ additions, so asympotically $2mnl$ elementary operations.

Rank and complexity

Matrix multiplication can be seen as a tensor

$$t_{m,n,l} \in M_{m,n} \otimes M_{n,l} \otimes M_{m,l}$$

$$t_{m,n,l}(A \otimes B \otimes C) = \sum_{i,j,k} a_{ik} b_{kj} c_{ji} = tr(ABC)$$

and we will see the number of multiplications needed coincides asymptotically with the rank of $t_{m,n,l}$.

The exponent ω of matrix multiplication.

Border rank$(t) = \mathrm{brk}(t) := \min\{r | \exists s_p \to t \text{ with } \mathrm{rk} s_p = r\}$, in more geometric terms

$$\mathrm{brk}(t) := \min\{r | t \in \sigma_r \left(\mathbb{P}(A) \times \mathbb{P}(B) \times \mathbb{P}(C)\right)\}$$

$\mathrm{brk}(t) \leq \mathrm{rk}(t)$, for d-way tensors with $d \geq 3$ there are examples where strict inequality holds.

The exponent of matrix multiplication ω is defined to be $\underline{\lim}_n \log_n$ of the arithmetic cost to multiply $n \times n$ matrices, or equivalently, $\underline{\lim}_n \log_n$ of the minimal number of multiplications needed.

Theorem (Strassen)

$$\omega = \underline{\lim}_n \log_n(\mathrm{brk} t_n)$$

Allowing approximations, the border rank of matrix mult. tensor t_n is a good measure of the complexity of matrix mult. algorithm .

Strassen result on 2×2 multiplication

Strassen showed explicitly

$$
\begin{aligned}
t_2 = t_{2,2,2} = {} & \underbrace{a_{11} \otimes b_{11} \otimes c_{11}}_{1} + \underbrace{a_{12} \otimes b_{21} \otimes c_{11}}_{2} + \underbrace{a_{21} \otimes b_{11} \otimes c_{21}}_{3} + \underbrace{a_{22} \otimes b_{21} \otimes c_{21}}_{4} \\
& + \underbrace{a_{11} \otimes b_{12} \otimes c_{12}}_{5} + \underbrace{a_{12} \otimes b_{22} \otimes c_{12}}_{6} + \underbrace{a_{21} \otimes b_{12} \otimes c_{22}}_{7} + \underbrace{a_{22} \otimes b_{22} \otimes c_{22}}_{8}
\end{aligned}
$$

$$
\begin{aligned}
= {} & \underbrace{(a_{11} + a_{22}) \otimes (b_{11} + b_{22}) \otimes (c_{11} + c_{22})}_{1} + \underbrace{(a_{21} + a_{22}) \otimes b_{11} \otimes (c_{21} - c_{22})}_{2} \\
& + \underbrace{a_{11} \otimes (b_{12} - b_{22}) \otimes (c_{12} + c_{22})}_{3} + \underbrace{a_{22} \otimes (-b_{11} + b_{21}) \otimes (c_{21} + c_{11})}_{4} \\
& + \underbrace{(a_{11} + a_{12}) \otimes b_{22} \otimes (-c_{11} + c_{12})}_{5} + \underbrace{(-a_{11} + a_{21}) \otimes (b_{11} + b_{12}) \otimes c_{22}}_{6} \\
& + \underbrace{(a_{12} - a_{22}) \otimes (b_{21} + b_{22}) \otimes c_{11}}_{7} .
\end{aligned}
$$

rank of 2×2 multiplication tensor

Theorem

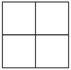
Rank and border rank of 2×2 *multiplication tensor are both* 7.

Theoretical proof by Landsberg (2006) with representation theory techniques.
Recent computational proof by Hauenstein, Ikenmeyer, Landsberg (2013).

In this case the rank of general tensor of the same size is again 7.
No more true in 3×3 case, where the rank and the border rank of multiplication tensor are both not known.

Iteration of Strassen result, analogy with Karatsuba

Dividing a matrix of size $2^k \times 2^k$ into 4 blocks of size $2^{k-1} \times 2^{k-1}$



one shows inductively that are needed 7^k multiplications and $9 \cdot 2^k$ additions, so a total of $9 \cdot 2^k + \underbrace{18 \cdot 7^{k-1}}_{\text{leading term}} \leq C7^k$ elementary operations.

The number 7 of multiplications needed turns out to be the crucial measure.

A consequence of Strassen bound is that $\omega \leq log_2 7 = 2.81\ldots$, which is better than $\omega \leq 3$ coming from the naive algorithm.

The rank and the border rank in case 3×3 are still unknown.

Comparison between polynomial multiplication and matrix multiplication

Complexity analysis

	polynomial multiplication	$n \times n$ matrix multiplication
$n = 2$	$3 < 4$ mult., Karatsuba	$7 < 8$ mult., Strassen
iterating	$n^{1.585\ldots}$ iterate Karatsuba	$n^{2.81}$ iterate Strassen
n	$O(n \log n)$ Fast Fourier Transf.	?
Rank	$2n+1$	?
BRank	$2n+1$	?

In the boxes marked "?" there are partial results.

The world record history for ω

ω=exponent of matrix multiplication

- Strassen, $O(n^{2.81})$, 1969
- Bini, Capovani, Romani, Lotti, $O(n^{2.7799})$, 1979
- Strassen, $O(n^{2.48})$, 1987, Laser method
- Coppersmith Vinogradov, $O(n^{2.375477})$, 1990
- Stothers, $O(n^{2.3736})$, 2010
- Williams, $O(n^{2.3729})$, 2012
- with different techniques, Davie, Stother, (Le Gall), $O(n^{2.37294})$, 2012

Cohn-Umans approach

Cohn-Umans made visible first the parallel between polynomial multiplication and matrix multiplication.
Let G be a finite group, the group algebra $\mathbb{C}[G]$ (regular representation of G) splits as

$$\mathbb{C}[G] = \oplus_i \mathrm{End}(W_i)$$

where W_i are the irreducible representations of G
Discrete Fourier Transform is the embedding of Polynomial multiplication into Group Algebra multiplication for the cyclic group G.
Cohn-Umans do the analogous embedding of Matrix Multiplication into Group Algebra Multiplication for non abelian groups of potential size $n^{2+o(1)}$. Actual bound on ω achieved with this technique is $O(n^{2.41})$. Is $\omega = 2$?

Bounds on border rank

Our results are as follows:

Theorem (Landsberg-O)

Let $n \leq m$.

$$brk(t_{m,n,l}) \geq \frac{nl\,(n+m-1)}{m}.$$

Corollary

$$brk(t_{n,n,l}) \geq 2nl - l$$

$$brk(t_n) \geq 2n^2 - n.$$

Thus for 3×3 matrices, the state of the art, up to 2013, is $15 \leq \mathrm{brk}(t_3) \leq 21$, the upper bound is due to Schönhage . Smirnov announced in 2014 that $\mathrm{brk}(t_3) \leq 20$.

3×3 case

Our computation started in the 3×3 case.
In this case the multiplication tensor sits in
$\mathbb{C}^9 \otimes \mathbb{C}^9 \otimes \mathbb{C}^9 = A \otimes B \otimes C \simeq \mathbb{C}^{729}$
We have a contraction map

$$t_3^{\wedge 4} \colon \mathbb{C}^{1134} \simeq \underbrace{B^\vee \otimes \wedge^4 A}_{\text{source space}} \to \underbrace{C \otimes \wedge^5 A}_{\text{target space}} \simeq \mathbb{C}^{1134}$$

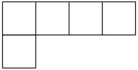
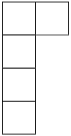
and the maximum rank expected is 1134.
The answer was $\mathrm{rk} t_3^{\wedge 4} = 918$. Note that, by replacing the multiplication tensor t_3 with a random tensor of the same format, the computer gives rank 1134.
Now $\mathrm{brk}(t_3) \geq \frac{\mathrm{rk}(t_3^4)}{\binom{8}{4}} = \frac{918}{\binom{8}{4}} = 13.11$
so it follows $\mathrm{brk}(t_3) \geq 14$, same than previously known Bläser bound. (Then we improved to 15, as we see in a while.)

Explanation of rank=918, through representation theory.

Indeed, write $A = M \otimes N^\vee$, $B = N \otimes L^\vee$, $C = L \otimes M^\vee$, so

$$B^\vee \otimes \wedge^4 A = L \otimes N^\vee \otimes \left(\oplus_{|\alpha|=4} S^\alpha M \otimes S^{\alpha'} N^\vee \right)$$

where α is a Young diagram and α' its transpose. Decomposing into irreducible summands, we see that $S^{2,1,1}M \otimes S^{4,1}N^\vee$ appears on the source space, but it cannot appear on the target space because the transpose of

 is  which has most of three rows and vanishes.

Hence the kernel is exactly
$L \otimes S^{2,1,1}M \otimes S^{4,1}N^*$ which has dimension $3 \cdot 3 \cdot 24 = 216$ and $1134 - 216 = 918$.

Idea for the proof

The essential idea for the proof is to choose a subspace $A' \subset M \otimes N^*$ on which the restriction of ${t_{m,n,l}}^{\wedge p}$ becomes injective. Take a vector space W of dimension 2, and fix isomorphisms $N \simeq S^{n-1}W$, $M \simeq S^{m-1}W^*$. Let A' be the direct summand $S^{m+n-2}W^* \subset S^{n-1}W^* \otimes S^{n-1}W^* = M \otimes N^*$, like in polynomial multiplication.
Recall that $S^\alpha W$ may be interpreted as the space of homogenous polynomials of degree α in two variables. If $f \in S^\alpha W$ and $g \in S^\beta W^*$ then we can perform the contraction $g \cdot f \in S^{\alpha-\beta}W$. In the case $f = l^\alpha$ is the power of a linear form l, then the contraction $g \cdot l^\alpha$ equals $l^{\alpha-\beta}$ multiplied by the value of g at the point l, so that (for $\beta \leq \alpha$) $g \cdot l^\alpha = 0$ if and only if l is a root of g.

The proof, I

Consider the $SL(2)$-equivariant skew-symmetrization map

$$A' \otimes \wedge^{n-1}(A') \longrightarrow \wedge^n (A')$$

Recall that representation theory distinguishes a complement A'' to A, so the projection $M \otimes N^\vee \longrightarrow A'$ is well defined. Compose with the projection

$$M \otimes N^\vee \otimes \wedge^{n-1}(A') \longrightarrow A' \otimes \wedge^{n-1}(A')$$

to obtain

$$M \otimes N^\vee \otimes \wedge^{n-1}(A') \longrightarrow \wedge^n (A').$$

and

$$\psi'_p : N^\vee \otimes \wedge^{n-1}(A') \longrightarrow M^\vee \otimes \wedge^n(A').$$

We claim it is injective. (Note that when $n = m$ the source and target space are dual to each other.)

The proof, II

Consider the transposed map $S^{m-1}W^{\vee} \otimes \wedge^{n} S^{m+n-2}W \longrightarrow S^{n-1}W \otimes \wedge^{n-1} S^{m+n-2}W$. It is defined as follows on decomposable elements (and then extended by linearity):

$$g \otimes (f_1 \wedge \cdots \wedge f_n) \mapsto \sum_{i=1}^{n} (-1)^{i-1} g(f_i) \otimes f_1 \wedge \cdots \hat{f_i} \cdots \wedge f_n$$

We show this dual map is surjective. Let $l^{n-1} \otimes (l_1^{m+n-2} \wedge \cdots \wedge l_{n-1}^{m+n-2}) \in S^{n-1}W \otimes \wedge^{n-1} S^{m+n-2}W$ with $l_i \in W$. Such elements span the target so it will be sufficient to show any such element is in the image. Assume first that l is distinct from the l_i. Since $n \leq m$, there is a polynomial $g \in S^{m-1}W^{\vee}$ which vanishes on $l_1, \ldots, l_{n-1}$ and is nonzero on l. Then, up to a nonzero scalar, $g \otimes (l_1^{m+n-2} \wedge \cdots \wedge l_{n-1}^{m+n-2} \wedge l^{m+n-2})$ maps to our element.

The proof, III

Since the image is closed (being a linear space), the condition that l is distinct from the l_i may be removed by taking limits.
Finally, $\psi'_p \otimes Id_L$ is the map induced from the restricted matrix multiplication tensor and we may repeat the general arguments.
To complete the proof, observe that an element of rank one in $A' \otimes B \otimes C$ induces a map of rank $\binom{n+m-2}{n-1}$. So the rank of the matrix multiplication tensor must be at least

$$\frac{\dim L \otimes N^\vee \otimes \wedge^{n-1}(A')}{\binom{n+m-2}{n-1}} = nl\frac{\binom{n+m-1}{n-1}}{\binom{n+m-2}{n-1}} = \frac{nl\,(n+m-1)}{m}.$$

which proves our result. It gives 15 for 3×3 multiplication (substitute $m = n = l = 3$).

Lower bounds on the rank

Bläser bound on the rank (1999)

Bläser proved

$$\mathrm{rk}(t_n) \geq \frac{5}{2}n^2 - 3n$$

Recent improvements

Following [Landsberg-O] technique, for every $p \leq n$

$$\mathrm{rk}(t_n) \geq (3 - \frac{1}{p+1})n^2 - h(p)n,$$

where $h(p) = (1 + 2p\binom{2p}{p})$, by [Landsberg, 2012] and then,

[Massarenti- Raviolo, 2013] improved the bound by replacing $h(p)$ with a smaller function of p.

Basic book references

Bürgisser, Clausen, Shokrollai, *Algebraic Complexity Theory*, **Springer, 1997**

J.M. Landsberg, *Tensors, Geometry and Applications*, **AMS, 2012**

Thanks for your attention!!