Electronic Colloquium on Computational Complexity, Comment 1 on Report No. 117 (2014)

Fast Approximate Matrix Multiplication in ECCC TR14-117 Fails to Converge

Kevin Kowalski* Dieter van Melkebeek*

February 6, 2015

Abstract

The paper [1] presents an efficient iterative algorithm that purportedly computes an approximation to the product of two $n \times n$ matrices. We show that, in fact, the algorithm converges to a different matrix than the product matrix.

1 Overview

In [1] the authors claim to have discovered an $O(n^2 \log \delta^{-1})$-time algorithm for approximately multiplying two $n \times n$ matrices where each entry is bounded by a constant such that the Frobenius distance between the returned product and the correct product is at most δ. The algorithm consists of three main steps:

1. a framing step, where the problem of matrix multiplication is reduced to solving a system of linear equations;
2. a perturbation step, where the system of linear equations is tweaked to be positive definite; and
3. an approximation step, where an approximate solution to the tweaked system is found.

The flaw in this algorithm is that the perturbation forces the approximation procedure to converge to an incorrect solution.

2 Summary of the algorithm

The problem of matrix multiplication can be stated as follows: given two $n \times n$ matrices A and B, return a matrix C such that $AB = C$. In this setting, the matrix C contains n^2 entries of unknown value, so the problem can be cast as solving a system of n^2 equations (each of the form $C_{ij} = A_{i\cdot} \cdot B_{\cdot j}$) over n^2 variables $\{C_{ij}\}_{i,j \in [n]}$.

In the algorithm of [1], the system of equations $AB = C$ is reduced to a new system as follows. Let v be a chosen $n \times 1$ vector, and consider the system $ABv = Cv$. This system can be rewritten as $ABv = Vc$ where V is an $n \times n^2$ block-diagonal matrix where each block is v^T, and c is an $n^2 \times 1$ variable vector where each block of n entries constitutes a row of C. More formally,

$$c = \texttt{Flatten}(C),$$

*Department of Computer Sciences, University of Wisconsin-Madison, Madison, WI 53706, USA. Partially supported by NSF grant CCF-1319822.

ISSN 1433-8092

where for any $n \times n$ matrix X, $x = \texttt{Flatten}(X)$ is the $n^2 \times 1$ vector such that $x_{n(i-1)+j} = X_{ij}$. The new system is then given by

$$V^T ABv = V^T Vc. \tag{1}$$

Note that this system is underdetermined–the $n^2 \times n^2$ matrix $V^T V$ has rank at most n, so the system has n^2 variables and at most n linearly independent equations. The vector $c = \texttt{Flatten}(AB)$ is one solution to the system, but there are necessarily many others.

Additionally, note that the system (1) is positive semidefinite–the coefficient matrix $V^T V$ is block-diagonal where each block is vv^T and the only nonzero eigenvalue of vv^T is $\sum_i v_i^2$. Hence, the eigenvalues of $V^T V$ are 0 with multiplicity $n^2 - n$ and $\sum_i v_i^2$ with multiplicity n, which are all nonnegative. The algorithm then perturbs $V^T V$ into a positive definite matrix by adding ϵI, where ϵ is a small constant and I is the identity matrix. This adds ϵ to all the eigenvalues of the coefficient matrix, so they all become positive. The perturbed system is then

$$V^T ABv = (V^T V + \epsilon I)c. \tag{2}$$

Since the perturbed system is positive definite, it has exactly one solution. A close approximation to this solution can be found efficiently using a steepest descent method.

3 Explanation of the flaw

The issue with the perturbed system (2) is that its unique solution, say c^*, when written as an $n \times n$ matrix instead of an $n^2 \times 1$ vector is $AB\left[vv^T/(v^T v + \epsilon)\right]$ rather than the correct product $C \doteq AB$. This fact is stated and proved in the following claim. Since vv^T is a rank 1 matrix, the solution also has rank at most 1, so it is impossible in nearly every case for a matrix close to the solution to also be close to the correct product.

Claim. *Let*

$$c^* \doteq \texttt{Flatten}(C\left[vv^T/(v^T v + \epsilon)\right]).$$

Then, $c = c^*$ is the unique solution to the perturbed system (2).

Proof. First note that by definition, for all $i, j \in [n]$, the $(n(i-1)+j)$-th entry of $V^T Cv$ is $v_j \sum_{k=1}^n C_{ik} v_k$. Additionally, for all $i, j \in n$, $c^*_{n(i-1)+j} = \sum_{k=1}^n C_{ik} v_k v_j/(v^T v + \epsilon)$. Hence, the $(n(i-1)+j)$-th entry of $(V^T V + \epsilon I)c^*$ is

$$\begin{aligned} \left[\sum_{\ell=1}^n v_j v_\ell c^*_{n(i-1)+\ell}\right] + \epsilon c^*_{n(i-1)+j} &= \frac{v_j}{v^T v + \epsilon}\left[\left(\sum_{\ell=1}^n v_\ell^2\right)\left(\sum_{k=1}^n C_{ik} v_k\right) + \epsilon \sum_{k=1}^n C_{ik} v_k\right] \\ &= v_j \sum_{k=1}^n C_{ik} v_k, \end{aligned}$$

so $V^T Cv = (V^T V + \epsilon I)c^*$ and $c = c^*$ is a solution to (2), as desired. $\square$

Given that in nearly every case the output of the algorithm is distant from the correct product, the error analysis in section 4.3 of [1], which claims otherwise, must have some issues. The heart of the analysis lies in Lemma 4, which attempts to bound the Frobenius distance between the output of the algorithm and "the" solution[1] to the unperturbed, positive semidefinite system (1), denoted by

[1] First sentence of section 4.3 in [1].

x''' in the paper. The incorrect implicit assumption that (1) has a unique solution that corresponds to the correct product, i.e., $x''' = \texttt{Flatten}(C)$, is used in the equality in the last line on page 7 of [1].

In addition, even if $x''' = \texttt{Flatten}(C)$, there are a number of errors in the derivation of an upper bound on the 2-norm of x''', denoted $|x'''|$. The analysis assumes values of $v = \begin{bmatrix} 1/n^3 & \cdots & 1/n^3 \end{bmatrix}^T$ and $\epsilon = 1/n^3$, and using these, claims that $|x'''| \leq n^{-5/2} \cdot \frac{\lambda+\epsilon}{\epsilon} \cdot M'$, where M' denotes the maximum row sum of any row in C and $1 \leq (\lambda + \epsilon)/\epsilon \leq 2$. Due to a mistake in parenthesization on the third line in the proof of Lemma 4 in [1], the actual bound obtained is a factor of n^3 larger, i.e., $|x'''| \leq \sqrt{n} \cdot \frac{\lambda+\epsilon}{\epsilon} \cdot M'$. Note that a better upper bound of $|x'''| \leq \sqrt{n} \cdot M'$ follows from the known relationships between the Frobenius norm and the row sum norm of a matrix.

Due to the additional factor, the upper bound on $|x'''|$ no longer vanishes when n grows. In fact, as x''' is supposed to be $\texttt{Flatten}(C)$, it is impossible to upper bound $|x'''|$ by a vanishing term, and the approach of showing convergence by bounding the error term $|x' - x'''|$ as $|x' - x'''| \leq |x' - x''| + |x''| + |x'''|$ cannot work.

References

[1] Shiva Manne and Manjish Pal. Fast approximate matrix multiplication by solving linear systems. *Electronic Colloquium on Computational Complexity (ECCC)*, TR14-117, 2014.

ECCC ISSN 1433-8092

http://eccc.hpi-web.de