Theoretical Computer Science 609 (2016) 76–86

Contents lists available at ScienceDirect

Theoretical Computer Science

www.elsevier.com/locate/tcs

Average-case complexity of the min-sum matrix product problem ☆

Ken C.K. Fong [a,*], Minming Li [a], Hongyu Liang [b], Linji Yang [c], Hao Yuan [d]

[a] *Department of Computer Science, City University of Hong Kong, Hong Kong*
[b] *Facebook, Inc., USA*
[c] *Georgia Institute of Technology, USA*
[d] *Bopu Technologies, China*

A R T I C L E   I N F O

Article history:
Received 12 January 2015
Received in revised form 17 July 2015
Accepted 6 September 2015
Available online 10 September 2015
Communicated by G.F. Italiano

Keywords:
Min-sum product
Min-plus product
Matrix multiplication

A B S T R A C T

We study the average-case complexity of min-sum product of matrices, which is a fundamental operation that has many applications in computer science. We focus on optimizing the number of "algebraic" operations (i.e., operations involving real numbers) used in the computation, since such operations are usually expensive in various environments. We present an algorithm that can compute the min-sum product of two $n \times n$ real matrices using only $O(n^2)$ algebraic operations, given that the matrix elements are drawn independently and identically from some fixed probability distribution satisfying several constraints. This improves the previously best known upper-bound of $O(n^2 \log n)$. The class of probability distributions under which our algorithm works include many important and commonly used distributions, such as uniform distributions, exponential distributions, folded normal distributions, etc.

In order to evaluate the performance of the proposed algorithm, we performed experiments to compare the running time of the proposed algorithm with algorithms in [1]. The experimental results demonstrate that our algorithm achieves significant performance improvement over the previous algorithms.

1. Introduction

The min-sum product (also known as min-plus product, distance matrix product, and distance matrix multiplication) of two matrices is defined as follows: Given two $n \times n$ real matrices A and B, the min-sum product of them, denoted by $A \otimes B$, is defined as a matrix C where

$$C_{i,j} \stackrel{\text{def}}{=} \min_k (A_{i,k} + B_{k,j}), \quad \text{for all } 1 \le i, j \le n. \tag{1}$$

The min-sum product is a fundamental operation that has many applications in computer science. For example, it is well known that the min-sum product problem is closely related to the problem of computing all-pairs shortest paths in a graph [2], which is among the most fundamental and well-studied problems in the algorithm community. Another

☆ This work was fully supported by a grant from the Research Grants Council, University Grants Committee, Hong Kong Special Administrative Region, China [Project No. CityU 122512].

* Corresponding author.
E-mail address: ken.fong@my.cityu.edu.hk (K.C.K. Fong).

http://dx.doi.org/10.1016/j.tcs.2015.09.008
0304-3975/

important application of the min-sum product is in performing MAP (maximum a posteriori) inference with graphical models [3,1].

A naïve implementation for computing the min-sum product of two $n \times n$ matrices requires $O(n^3)$ time, which is too slow for a large n. Because of the importance of this problem, many algorithms were developed to improve the cubic time bound (see the work of Chan [4] and the references therein). Currently, the best-known worst-case algorithm is due to the result of William [5] with time complexity of $O(n^3/2^{\Omega(\log n)^{1/2}})$ which improves the result of Han and Takaoka [6] on the all pairs shortest paths. This is only slightly better than cubic time by a poly-logarithmic factor. Whether there exists a truly worst-case sub-cubic (i.e., $O(n^{3-\delta})$ for some positive constant δ) algorithm for the min-sum product problem is a long-standing open problem. The difficulty comes from the fact that the min-sum product is computed on a semiring structure, where there is no additive inverse defined over the min operator. Existing truly sub-cubic fast matrix multiplication algorithms (e.g., the Strassen algorithm [7] and the Coppersmith–Winograd algorithm [8]) only work on a ring structure. In fact, it is conjectured by Chan that an $n^{3-\Omega(1)}$ time algorithm does not exist for the min-sum product problem [4]. The authors in [9,6] also agreed with this conjecture.

In many practical applications of the min-sum product problem, the average-case time complexity becomes more interesting than worst-case complexity. As shown in a recent work of Felzenszwalb and McAuley [1], the min-sum product problem can be solved significantly faster than cubic time in the average case for some MAP inference applications in computer vision and natural language processing. (See also the work of McAuley and Caetano [3] for the related applications.) The algorithm of Felzenszwalb and McAuley [1] runs in expected $O(n^2 \log n)$ time when the entries of the input matrices are independently drawn from a uniform distribution on $[0, 1]$. A more general algorithm of Takaoka [10] also runs in expected $O(n^2 \log n)$ time under the endpoint independence model [11]. Although being almost tight, there is still an $O(\log n)$ gap between the upper bound and the trivial lower bound of $\Omega(n^2)$ (which is the time required to output the answer).

Since the computations involving real numbers are usually much costlier than that of integers, in real applications it is useful to consider the following restricted algebraic model of computation: Computations involving real numbers are restricted to adding two real numbers and comparing two real numbers. When talking about the *algebraic complexity*, we only count the costs for adding and comparing two real numbers; all other computations (like adding two integer variables, comparing two indices, etc.) are assumed to be cost-free.

For the min-sum product problem, this restricted computation model actually coincides with the decision tree model [12], which also counts only the number of comparisons and additions of the matrix elements (or edge weights in the graph view) required in the computation. Thus, the result of [13] gives an $O(n^{2.5})$ worst-case bound on the algebraic complexity of min-sum product, which is a substantial improvement on the $n^{3-o(1)}$ complexity in the traditional model. However, this is not the case when considering the average case. The analyses of previous algorithms [10,1] only give an $O(n^2 \log n)$ bound on the average-case algebraic complexity, which is of the same order with the traditional average-case complexity. It is not clear from the previous studies whether this bound can be further improved.

Our contributions In this paper, we study the problem of computing the min-sum product of two random matrices under the aforementioned algebraic computation model. We show that the min-sum product of two $n \times n$ matrices can be computed using only $O(n^2)$ expected algebraic operations (i.e., adding or comparing two real numbers), when the elements of the two matrices are independently drawn from some fixed probability distribution satisfying several constraints (see Theorem 1). Thus, our result improves the algebraic complexity of computing the min-sum product of two matrices from $O(n^2 \log n)$ [10,1] to $O(n^2)$, which is clearly the best possible due to the trivial quadratic lower bound. The class of probability distributions under which our algorithm works include many important and commonly used distributions, e.g., the uniform distribution on $[0, \vartheta]$ for any $\vartheta > 0$, exponential distributions, folded normal distributions, etc. As mentioned before, the algebraic computation model actually coincides with the decision tree model. Therefore, as a by-product, the average-case decision tree complexity of min-sum product is shown to be $O(n^2)$, which matches the trivial $\Omega(n^2)$ lower bound.

Besides the theoretical result we achieved, we re-implemented the algorithms in [1] and our algorithm in C++ to perform the comparison on the running time. For two matrices multiplication, the experimental results show that our algorithm achieves significant performance improvements over the previous algorithms, especially when n is large. Moreover, we also conducted the experiments with multiple matrices multiplication. Fig. 2 shows that the improvement over algorithms in [1] is significant when m is small.

We note that a recent work of Peres et al. [14] solves the all-pairs shortest paths problem on a complete graph (or the $G(n, p)$ model with moderately large p) in expected $O(n^2)$ time when the edge lengths are from the uniform distribution on $[0, \vartheta]$. Their work improves several previous results on the average-case complexity of the problem (e.g., [11,15–19]). However, neither their algorithm nor the previous ones apply to our case since they only work for complete random graphs.

2. Min-sum product of two matrices

Let A and B be two n by n matrices, and $C = A \otimes B$ be their min-sum product. Assume that the entries of A and B are i.i.d. random variables drawn from some fixed probability distribution. Our goal is to efficiently compute C.

As introduced earlier, we focus on a restricted algebraic model of computation as follows: Computations that involve real numbers are restricted to adding two real numbers and comparing two real numbers; no other computations are