arXiv:2603.19443v1 [cs.DS] 19 Mar 2026

惰性克罗内克积

Zhao Song*

摘要

本文展示了如何将动态矩阵乘积中的惰性更新机制[Cohen、Lee、Song，STOC 2019，JACM 2021] [CLS21] 推广到动态克罗内克积。我们提出了一种算法，用于 $n^{\omega(\lceil k/2\rceil,\lfloor k/2\rfloor,a)-a}$ 实现摊还更新时间为 $n^{\omega(\lceil (k-s)/2\rceil,\lfloor (k-s)/2\rfloor,a)}$且最坏情况查询时间为 $n^{\omega(\lceil k/2\rceil,\lfloor k/2\rfloor,a)-a-\Omega(1)}$ 的动态克罗内克积问题。除非张量矩阵-向量猜想不成立，否则不存在一种算法能够同时实现 $n^{\omega(\lceil (k-s)/2\rceil,\lfloor (k-s)/2\rfloor,a)-\Omega(1)}$ 摊还更新时间为，且最坏情况查询时间为。

1 引言

我们定义动态克罗内克积问题：在每次迭代中，我们接收一个秩一更新和一个多维查询集。这可以看作是对 [CLS21] 中所介绍的矩阵版本的推广。考虑一个 k 阶张量 $A \in \mathbb{R}^{n\times\cdots\times n}$ 、一个正整数 s、一个索引集 $\{\ell_1,\cdots,\ell_s\}\subset[k]$ 以及一个多重集 $i=\{i_1,\cdots,i_s\}$，其中每个元素均取自 $[n]$。这里，$A_{\cdots,i_1,\cdots,i_s,\cdots}$ 表示 A 的一个子张量，它在 ℓ_1 方向上选取 i_1，在 ℓ_2 方向上选取 i_2，并在 ℓ_s 方向上选取 i_s。

定义 *1.1*（动态克罗内克积）。假设给定向量 $\{u_1(1)\cdots、u_1(T)\}\subset\mathbb{R}^n$、$\{u_2(1)\cdots、u_2(T)\}\subset\mathbb{R}^n$、$\cdots$、$\{u_k(1)\cdots、u_k(T)\}\subset\mathbb{R}^n$。令 s 表示一个正整数。在每次迭代 t 中，我们将接收 $(u_1(t)\ u_2(t)\cdots、u_k(t)\ \ell(t)\ i(t))$。这里，$\ell=\{\ell_1,\cdots,\ell_s\}$ 是一个索引集，它是 $[k]$ 的子集。而 $i=\{i_1,i_2,\cdots,i_s\}$ 是一个多重索引集，其中每个元素均取自 $[n]$。目标是在每次迭代 $t\in[T]$ 中输出答案 $(\sum_{i=1}^t\otimes_{j=1}^k u_j(i))_{\cdots、i_1、\cdots、i_s、\cdots}$。我们注意到，该输出是一个子张量。

我们的主要结果如下。

定理 1.2 (我们的结果，算法)。令 $k\geq s\geq 1$ 为正整数。对于任意 $a>0$，存在一种算法，其摊还更新时间为 $O(n^{\omega(\lceil k/2\rceil,\lfloor k/2\rfloor,a)-a})$，最坏情况查询时间为 $O(n^{\omega(\lceil (k-s)/2\rceil,\lfloor (k-s)/2\rfloor,a)})$。

证明。令 $K:=n^a$。不失一般性，假设 K 是一个正整数。我们维护一个张量 A 和低秩分量 $U_1,U_2,\cdots,U_k\in\mathbb{R}^{n\times K}$。每经过 K 次迭代，我们就通过计算 $A\leftarrow A+U_1\otimes U_2\otimes\cdots\otimes U_k$来更新 A。每次迭代的查询时间分为两部分：完整部分和低秩部分。(1) 完整部分。在这部分中，我们计算 $A_{\cdots,i_1,\cdots,i_s,\cdots}$，耗时 $O(n^{k-s})$。(2) 低秩部分。在这部分中，我们考虑计算 $(U_1\otimes U_2\otimes\cdots\otimes U_k)_{\cdots,i_1,\cdots,i_s,\cdots}=U_1\otimes\cdots\otimes v_{l_1}\otimes\cdots\otimes v_{l_s}\otimes\cdots\otimes U_k$所需的时间。这里，我们将张量 $s,U_{\ell_1},\cdots,U_{\ell_s}$ 替换为向量 s，$v_{\ell_1},\cdots,v_{\ell_s}$。令 w' 表示由 $\odot_{i=1}^s v_{\ell_i}$构造的向量。操作 $\odot$ 表示哈达玛积。令 W' 表示一个未被替换的矩阵。

*magic.linuxkde@gmail.com。

我们通过计算 $\mathrm{diag}(w')W'$来构造 V_1。令 $V_2, \cdots, V_{k-s}$ 表示其余未被替换的矩阵。我们计算 $B = \oslash_{j=1}^{\lceil (k-s)/2 \rceil} V_j$.[1] 类似地，计算 $C = \oslash_{j=\lceil (k-s)/2 \rceil+1}^{k-s} V_j$。因此，低秩部分总共耗时 $n^{\omega(\lceil (k-s)/2 \rceil, \lfloor (k-s)/2 \rfloor, a)}$。每经过 K 次迭代，我们需要更新 A。此更新耗时 $n^{\omega(\lceil k/2 \rceil, \lfloor k/2 \rfloor, a)}$。为说明这一点，我们将针对 $U_1, \cdots, U_k \in \mathbb{R}^{n \times K}$ 计算 $\otimes_{j=1}^{k} U_j$的过程分解为两步。首先，构造一个中间矩阵$B = \oslash_{j=1}^{\lceil k/2 \rceil} U_j$ 和 $C = \oslash_{j=\lceil k/2 \rceil+1}^{k} U_j$。接着，计算 $B \cdot C^\top$，耗时 $n^{\omega(\lceil k/2 \rceil, \lfloor k/2 \rfloor, a)}$（矩阵 $n^{\lceil k/2 \rceil} \times n^{\lfloor k/2 \rfloor}$ 与一个 k阶张量 $n \times n \times \cdots \times n$之间存在自然的一一对应关系）。因此，更新 A 所需的时间主要由第二步决定，所以每次更新的总耗时为 $n^{\omega(\lceil k/2 \rceil, \lfloor k/2 \rfloor, a)}$。由于每隔 K 次迭代执行一次更新，摊还更新时间为 $n^{\omega(\lceil k/2 \rceil, \lfloor k/2 \rfloor, a)-a}$。因此，摊还更新时间为 $O(n^{\omega(\lceil k/2 \rceil, \lfloor k/2 \rfloor, a)-a})$，最坏情况查询时间为$O(n^{\omega(\lceil (k-s)/2 \rceil, \lfloor (k-s)/2 \rfloor, a)})$。

□

我们注意到，对于 $k = 2$ 和 $s = 1$，运行时间退化为引理 5.4 中的更新时间与 [CLS21] 中引理 4.5 的查询时间部分。

[BNS19] 提出了带提示的矩阵向量乘积猜想，而 [Son26] 将其推广到了张量。

定义 1.3（带提示的矩阵向量乘积的张量版本，[Son26]）。在布尔半环上，对于固定参数 $\tau > 0$，我们将张量提示矩阵向量乘法问题定义为一个三阶段过程：阶段 *1*：接收 k 个矩阵 V_1、V_2、……、V_k，每个矩阵的大小为 $n \times d$。阶段 *2*：接收一个阶为 k 的对角张量P，其维度为 $d \times d \times \cdots \times d$，且最多包含 n^τ 个非零项。阶段 *3*：接收一组目标模式 $\{\ell_1, \ldots, \ell_s\} \subseteq [k]$ 以及从$[n]$中抽取的相应指标多重集 $\{i_1, \ldots, i_s\}$。目标是输出子张量 $[P\ (V_1、\ V_2、\ \ldots\ldots、\ V_k)\ ]_{\ldots,\ i_1\ldots\ldots、\ i_s\ldots\ldots}$，其中，对于所有 $t \in [k]$，每个指标 i_t 都将张量限制在相应的第 ℓ_t 个方向上。

猜想 1.4 ([Son26])。**对于任何在阶段 1 使用多项式预处理来解决张量提示矩阵向量乘法问题（定义 *1.3*）的算法，以下下界至少有一个必须成立：阶段 *2* 需要 $\Omega(n^{\omega(\lceil k/2 \rceil, \lfloor k/2 \rfloor, \tau)-\delta})$ 时间；阶段 *3* 对每个 $\delta > 0$，都需要 $\Omega(n^{\omega(\lceil (k-s)/2 \rceil, \lfloor (k-s)/2 \rfloor, \tau)-\delta})$时间。**

定理 1.5 (我们的结果：困难性)。**除非张量矩阵-向量猜想（猜想 *1.4*）不成立，否则不存在能够同时满足以下条件的算法：使用 $n^{\omega(\lceil k/2 \rceil, \lfloor k/2 \rfloor, a)-a-\delta}$ 摊销更新时间，以及 $n^{\omega(\lceil (k-s)/2 \rceil, \lfloor (k-s)/2 \rfloor, a)-\delta}$最坏查询时间，其中某个常数为 $\delta > 0$。**

证明。我们将通过归约来证明这一点。为导出矛盾，假设存在一个用于动态张量乘法问题（定义 1.1）的动态算法 $\mathcal{A}$，同时满足：摊还更新时间：$O(n^{\omega(\lceil k/2 \rceil, \lfloor k/2 \rfloor, a)-a-\delta})$，以及最坏情况查询时间：$O(n^{\omega(\lceil (k-s)/2 \rceil, \lfloor (k-s)/2 \rfloor, a)-\delta})$。我们将展示如何利用算法 $\mathcal{A}$，以快于猜想 1.4 所规定下界的速度解决张量提示矩阵向量乘法问题（定义 1.3），从而推翻该猜想。为使参数相匹配，我们设定 $a = \tau$。在张量提示矩阵向量乘法问题的阶段 1 中，我们获得 k 个矩阵 V_1、V_2、……、V_k，每个矩阵的大小为 $n \times d$。我们使用算法 $\mathcal{A}$ 初始化一个空的动态张量结构。该阶段充当我们的多项式预处理。在阶段 2 中，我们获得一个大小为 $d \times d \times \cdots \times d$ 的对角张量 P，其中至多有 n^τ 个非零项。设非零项的数量为 $m \leq n^\tau$。每个非零项 P_j、j、……、j 都对应一次秩一更新。对于这 m 个非零项中的每一个，我们都向动态算法 $\mathcal{A}$ 发出一次更新。更新向量就是给定矩阵的 j 列：$u_1(t) = (V_1)_{*,\ j}$、$u_2(t) = (V_2)_{*,\ j}$、……、$u_k(t) = (V_k)_{*,\ j}$。由于此类更新至多有 n^τ 次，且 $\mathcal{A}$ 的摊还更新时间为 $O(n^{\omega(\lceil k/2 \rceil, \lfloor k/2 \rfloor, \tau)-\tau-\delta})$，处理所有更新所需的总时间有界为

[1]给定矩阵 $A_1 \in \mathbb{R}^{n \times d}, A_2 \in \mathbb{R}^{n \times d}, \cdots A_k \in \mathbb{R}^{n \times d}$，我们将 $n^k \times d$矩阵 $B := \oslash_{j=1}^{k} A_j$定义为：其 $(p_1, \cdots, p_k)$ q项为 $\prod_{j=1}^{k} (A_j)_{p_j, q}$。

：$m \times O(n^{\omega(\lceil k/2 \rceil, \lfloor k/2 \rfloor, \tau) - \tau - \delta}) \leq n^{\tau} \times O(n^{\omega(\lceil k/2 \rceil, \lfloor k/2 \rfloor, \tau) - \tau - \delta}) = O(n^{\omega(\lceil k/2 \rceil, \lfloor k/2 \rfloor, \tau) - \delta})$。在阶段 3 中，我们获得一个索引集 $\{\ell_1, \ldots, \ell_s\} \subseteq [k]$ 和一个指标多重集 $\{i_1, \ldots, i_s\}$。我们需要计算子张量 $[P(V_1、\ldots\ldots、V_k)]_{\ldots、i_1、\ldots\ldots、i_s、\ldots\ldots}$。这正是动态张量乘法问题（定义 1.1）中所定义的查询操作。我们使用这些指标查询算法 $\mathcal{A}$。根据我们的假设，算法 $\mathcal{A}$ 的最坏情况查询时间为：$O(n^{\omega(\lceil (k-s)/2 \rceil, \lfloor (k-s)/2 \rfloor, \tau) - \delta})$。由此，我们构造出了一个解决张量提示矩阵向量乘法问题的求解器，其运行时间如下：阶段 2 时间：$O(n^{\omega(\lceil k/2 \rceil, \lfloor k/2 \rfloor, \tau) - \delta})$。阶段 3 时间：$O(n^{\omega(\lceil (k-s)/2 \rceil, \lfloor (k-s)/2 \rfloor, \tau) - \delta})$。然而，猜想 1.4 声称，对于任意算法，对每个 $\delta > 0$，以下两项中至少有一项必须成立：阶段 2 需要 $\Omega(n^{\omega(\lceil k/2 \rceil, \lfloor k/2 \rfloor, \tau) - \delta})$。阶段 3 需要 $\Omega(n^{\omega(\lceil (k-s)/2 \rceil, \lfloor (k-s)/2 \rfloor, \tau) - \delta})$。由于我们的算法 $\mathcal{A}$ 同时严格优于这两个下界（快了 n^{δ} 倍），我们得到了矛盾。因此，除非张量提示矩阵向量乘法猜想为假，否则这样的算法 $\mathcal{A}$ 不可能存在。

□

我们指出，上述证明与 [BNS19]中的定理 5.3 的证明类似。

参考资料

[BNS19] Jan van den Brand、Danupon Nanongkai 和 Thatchaphol Saranurak。动态矩阵求逆：改进的算法和相匹配的条件下界。发表于 *2019* 年 *IEEE* 第 *60* 届计算机科学基础年会（*FOCS*），第 456–480 页。IEEE，2019 年。[CLS21] Michael B Cohen、Yin Tat Lee 和 Zhao Song。在当前矩阵乘法时间内求解线性规划。《*ACM* 汇刊》（*JACM*），68(1)：1–39，2021 年。[Son26] Zhao Song。张量提示矩阵向量乘法猜想。*arXiv* 预印本 *arXiv:2602.07242*，2026 年。