fast matrix multiplication on large sparse matrices

韩海前

November 20, 2023

1 Introduction

大型稀疏矩阵一般是指维度较高，但是大部分元素均为0的矩阵。其数学定义如下

$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix}$$

$$\mathbf{A_s} = \{s_{ij} : a_{ij} \neq 0, i = 1, 2, \ldots, m, j = 1, 2, \ldots, n, s_i j \ll n^2\}$$

这样的矩阵并不少见，比如在图论中经常需要用邻接矩阵来表示图中的边：

$$A_{ij} = \begin{cases} 1, & \text{if there is an edge from vertex } i \text{ to vertex } j \\ 0, & \text{otherwise} \end{cases}$$

而邻接矩阵的乘法可以计算是否在两点间存在长度为2的通路。对于常见的，边数远小于点数的图，形成的邻接矩阵便是大型稀疏矩阵。 大型稀疏矩阵乘法在许多应用中都有广泛的应用，包括图像处理、信号处理、机器学习和科学计算。例如，在图像处理中，稀疏矩阵乘法可以用于图像去噪、图像分割和图像复原。在信号处理中，稀疏矩阵乘法可以用于信号压缩、信号分离和信号检测。在机器学习中，稀疏矩阵乘法可以用于特征提取、模型训练和模型评估。在科学计算中，稀疏矩阵乘法可以用于求解线性方程组、求解偏微分方程和求解大规模优化问题。

两个$n \times n$矩阵的乘法是最基本的代数问题之一，为此付出了相当大的努力以获得该任务的有效算法。简单的矩阵乘法算法的时间复杂度是$O(n^3)$。然而，对于大型稀疏矩阵来讲，这样的计算存在极大的浪费，因为大部分元素都是0，实际上会执行许多0乘以0的无效计算，这就给了算法优化可能的空间。

Algorithm *SMP*(A, B)

Input: Two $n \times n$ matrices A and B.
Output: The product AB.

1. Let a_k be the number of non-zero elements in A_{*k}, the k-th column of A, for $1 \leq k \leq n$.
2. Let b_k be the number of non-zero elements in B_{k*}, the k-th row of B, for $1 \leq k \leq n$.
3. Let π be a permutation for which $a_{\pi(1)}b_{\pi(1)} \geq a_{\pi(2)}b_{\pi(2)} \geq \ldots \geq a_{\pi(n)}b_{\pi(n)}$.
4. Find an $0 \leq \ell \leq n$ that minimizes $M(n, \ell, n) + \sum_{k>\ell} a_{\pi(k)}b_{\pi(k)}$.
5. Let $I = \{\pi(1), \ldots, \pi(\ell)\}$ and $J = \{\pi(\ell+1), \ldots, \pi(n)\}$.
6. Compute $C_1 \leftarrow A_{*I}B_{I*}$ using the fast dense rectangular matrix multiplication algorithm.
7. Compute $C_2 \leftarrow A_{*J}B_{J*}$ using the naive sparse matrix multiplication algorithm.
8. Output $C_1 + C_2$.

Figure 1: SMP

2 Methods

大型稀疏矩阵乘法有很多可能的办法[YZ05]（此处用中文似乎会导致引用乱码），此处首先介绍一种常见的方法,即图中的SMP方法（引用量比较高，有300多）： 上述算法的时间复杂度与具体的矩阵输入有关。在最好的情况下甚至可以是$O(n)$。因此对比一般的矩阵乘法还是有很大提升的。

3 Requirement

本课题准备如下的目标，最后会根据实际的推进情况和难度决定完成度。目标1和目标2是必须完成的基础部分，进阶目标顺序不分前后。

目标1：理解SMP算法的算法过程及其时间复杂度推导,完成对其的简述

目标2：完成SMP算法的代码，同时要测试其Strassen算法的时间复杂度对比（Strassen算法是对常规矩阵的一种优化，并不假设输入是稀疏矩阵，因此理论上在稀疏矩阵上表现较差，但在一般矩阵上可能反超SMP)

进阶目标3:我一直认为，对大型稀疏矩阵进行SVD分解后，分解得到的矩阵维度可能很低，主要信息可能主要在前几维度。因此可以研究利用SVD分解进行大型稀疏矩阵计算的可能性，并于SMP算法进行对比。

进阶目标4：随着GPU的发展，GPU上的平行计算大大促进了近几年计算科学的发展。但为了适应GPU的特殊结构，往往要为其设计全新的并行算法。可以尝试复现一种基于GPU的大型稀疏矩阵计算，并对比其与SMP的效果

References

[YZ05] Raphael Yuster and Uri Zwick. Fast sparse matrix multiplication. *ACM Transactions On Algorithms (TALG)*, 1(1):2–13, 2005.