Abstract

Let $A \in \{0,1\}^{m \times n}$ be a matrix with t_0 zeroes and t_1 ones and $\mathbf{x}$ be an n-dimensional vector over a semigroup. How many semigroup operations are required to compute $A\mathbf{x}$? This problem generalizes the well-known range queries problem and has applications in graph algorithms, functional programming languages, circuit complexity, and others. It is immediate that $O(t_1 + n + m)$ semigroup operations are sufficient. The main question studied in this paper is: can $A\mathbf{x}$ be computed using $O(t_0 + n + m)$ semiring operations? We prove that in general this is not possible: there exists a matrix $A \in \{0,1\}^{n \times n}$ having exactly two zeroes in every row (hence $t_0 = 2n$) whose complexity is $\Theta(n\alpha(n))$. However, for the case when the underlying semiring is commutative, we prove an $O(t_0+n+m)$ upper bound. This implies that for commutative setting, complements of sparse matrices can be processed as efficiently as spares matrices (though the corresponding algorithm is more involved).

Contents

1 Introduction

1.1 Problem Statement and New Results

Let $A \in \{0,1\}^{m \times n}$ be a matrix with t_0 zeroes and t_1 ones and $\mathbf{x} = (x_1, \dots, x_n)$ be an n-dimensional vector over a semigroup $(S, \circ)$. How many semigroup operations are required to compute the *linear operator* $A\mathbf{x}$? In this case, the i-th element of the output vector is $\sum_{j\colon A_{ij}=1} x_j$ where the summation is over the semigroup operation $\circ$. More generally, how many semigroup operations are needed to compute AB where $B \in S^{n \times k}$? In this paper, we are interested in lower and upper bounds involving t_0 and t_1. Matrices with $t_1 = O(n)$ are usually called *dense* whereas matrices with $t_2 = O(n)$ are called *complements of dense matrices*. It is not difficult to see that computing all n outputs of $A\mathbf{x}$ independently takes $O(t_1 + n + m)$ semigroup operations. The main question studied in this paper is: can $A\mathbf{x}$ be computed using $O(t_0 + n + m)$ semiring operations? (This complexity is easy to achieve if $\circ$ has an easily computable inverse (in that case S is a group): in this case A can be obtained by subtracting a dense matrix from all-ones matrix.) Our first result states that this is possible for *commutative* semigroups.

Theorem 1.1. *Let S be a commutative semigroup, $\mathbf{x} \in S^{n \times 1}$ be a vector, and $A \in \{0,1\}^{m \times n}$ be a matrix with t_0 zeros. Then, $A\mathbf{x}$ can be computed using at most $O(m+n+t_0)$ semiring operations.*

As an immediate consequence we get the following matrix multiplication result.

Corollary 1.2. *Let S be a commutative semigroup, $A \in \{0,1\}^{n \times n}$ be a matrix with t_0 zeros, and $B \in S^{n \times n}$ be a matrix. Then, AB can be computed using at most $O(n^2)$ semiring operations.*

We then show that commutativity is essential: for a strongly non-commutative semigroup S (the notion of strongly non-commutativity is made formal later in the text) the minimum number of semigroup operations needed to compute $A\mathbf{x}$ for a matrix $A \in \{0,1\}^{n \times n}$ with $t_0 = O(n)$ zeros is $\Theta(n\alpha(n))$ where is the inverse Ackermann function.

Lemma 1.3. *For any strongly non-commutative semigroup X there is a circuit to compute any dense operator of size $O(n\alpha(n))$, where $\alpha(n)$ is the inverse Ackermann function. On the other hand, there exist dense matrices A such that any circuit computing Ax must have size $\Omega(n\alpha(n))$.*

1.2 Motivation

The linear operator problem is interesting for many reasons.

Range queries. In the *range queries* problem, one is given a vector $\mathbf{x} = (x_1, \dots, x_n)$ over a semiring $(S, \circ)$ and a bunch of queries of the form (l, r) and is required to output the result $x_l \circ x_{l+1} \circ \cdots \circ x_r$ for each query. The linear operator problem is thus a natural generalization of the range queries problem: each row of the matrix A defines a subset of the elements of $\mathbf{x}$ that need to be summed up and this subset is not required to be a contiguous interval. The algorithms and hardness results for the linear operator problem presented in this paper are indeed inspired by some of the known results for the range queries problem. Later in the text we summarize a rich variety of algorithmic approaches and applications of the range queries problem.

Graph algorithms. Various graph path/reachability problems can be reduced naturally to matrix multiplication. Say, the all-pairs shortest path problem (APSP) is reducible to min-plus matrix product. Another example: the number of triangles in an undirected graph is equal to the trace of A^3 divided by six, where A is the adjacency matrix and matrix multiplication is over integers. It is natural to ask what happens if a graph has $O(n)$ edges or $O(n)$ anti-edges (as usual, by n we denote the number of nodes). In many cases, an efficient algorithm for sparse graphs is almost immediate whereas an algorithm with the same efficiency for a complement of a dense graph is not so immediate. For example, it is easy to solve APSP and triangle counting on dense graphs in time $O(n^2)$, but it is not immediate how to achieve the same time complexity for complements of dense graphs. In this paper, we show that the same effect occurs for linear operators: we prove that complements of dense matrices can be processed as efficiently as dense matrices though the corresponding algorithm is more involved. In particular, the efficient algorithm for matrix multiplication from Theorem ...

Matrix multiplication over semirings. Fast matrix multiplication methods rely essentially on the ring structure. The first such algorithm was given by Strassen [], the current record upper bound is $O(n^{2.373})$ []; see the survey [] for an overview of known approaches. For various important semirings lacking the inverse operation, we still do not know an $n^{3-\varepsilon}$ upper bound for a constant $\varepsilon > 0$. E.g., the strongest known upper bound for min-plus matrix multiplication is $n^3/\exp(\sqrt{\log n})$. In this paper, we present natural special cases of matrices where faster multiplication is possible.

For any strongly non-commutative semigroup X there is a circuit to compute any dense operator of size O

Functional programming languages **Circuit complexity.** Computing linear operators over a Boolean semiring $(\{0,1\}, \vee)$ is a well-studied problem in circuit complexity. The corresponding computational model is known as *rectifier network*. An overview of known lower and upper bounds for such circuits is given by Jukna [**?**, Section 13.6].

1.3 Organization of the Paper

For an