Complexity of Linear Operators

Vladimir V. Podolskii[1]

joint work with Alexander Kulikov, Ivan Mikhailin, Andrey Mokhov

[1] Steklov Mathematical Institute, Moscow
Higher School of Economics, Moscow

ISAAC 2019

Setting

Consider a Boolean matrix $A \in \{0,1\}^{n \times n}$
Consider variables $x = (x_1, \ldots, x_n)$ over $\{0,1\}$

We want to compute a Boolean linear operator Ax

$$A = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix} \qquad Ax = \begin{pmatrix} x_1 \vee x_3 \\ x_1 \vee x_2 \\ x_2 \vee x_3 \vee x_4 \\ x_1 \vee x_2 \vee x_3 \vee x_4 \end{pmatrix}$$

The Model

- The computation is a Boolean circuit consisting of OR gates
- We start with variables $x_1, \ldots, x_n$
- In one step we can compute OR of two previously computed expressions
- Want to compute all the outputs and minimize the number of steps

Example

$$A = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix}$$

$x_1 \vee x_3$
$x_1 \vee x_2$
$x_2 \vee x_3 \vee x_4$
$x_1 \vee x_2 \vee x_3 \vee x_4$

Computation:

$x_1 \vee x_3$ — output
$x_1 \vee x_2$ — output
$x_3 \vee x_4$
$x_2 \vee (x_3 \vee x_4)$ — output
$x_1 \vee (x_2 \vee x_3 \vee x_4)$ — output

Basic facts

- One of the simplest Boolean circuit complexity models, studied since 50's
- Trivial upper bound: $O(n^2)$
- Counting lower bound: $\Omega(n^2/\log n)$
- Non-trivial upper bound: $O(n^2/\log n)$ (Lupanov '56)
- The best explicit lower bound: $\Omega(n^{2-o(1)})$ (Nechiporuk '70)

General setting

Consider a Boolean matrix $A \in \{0,1\}^{n \times n}$
Consider variables $x = (x_1, \ldots, x_n)$ over some semigroup $(S, \circ)$.

We want to compute a linear operator Ax.

$$A = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix} \qquad Ax = \begin{pmatrix} x_1 \circ x_3 \\ x_1 \circ x_2 \\ x_2 \circ x_3 \circ x_4 \\ x_1 \circ x_2 \circ x_3 \circ x_4 \end{pmatrix}$$

Some semigroups

- Boolean semigroup: $(\{0,1\}, \vee)$
- Integers with addition: $(\mathbb{Z}, +)$
- $\{0,1\}$ with addition: $(\{0,1\}, \oplus)$
- Tropical semigroup: $(\mathbb{Z}, \min)$

The Problem

Simplified Problem: Suppose $A \in \{0,1\}^{n \times n}$ is very dense, that is A has $O(n)$ zeros. How hard is it to compute Ax?

The Problem

Simplified Problem: Suppose $A \in \{0,1\}^{n \times n}$ is very dense, that is A has $O(n)$ zeros. How hard is it to compute Ax?

Simple observations:

- If instead we consider A containing $O(n)$ ones, the complexity is trivially $O(n)$

The Problem

Simplified Problem: Suppose $A \in \{0,1\}^{n \times n}$ is very dense, that is A has $O(n)$ zeros. How hard is it to compute Ax?

Simple observations:

- If instead we consider A containing $O(n)$ ones, the complexity is trivially $O(n)$
- If $(S, \circ)$ has an inverse operation (is a group), the complexity is trivially $O(n)$

The Problem

Simplified Problem: Suppose $A \in \{0,1\}^{n \times n}$ is very dense, that is A has $O(n)$ zeros. How hard is it to compute Ax?

Simple observations:

- If instead we consider A containing $O(n)$ ones, the complexity is trivially $O(n)$
- If $(S, \circ)$ has an inverse operation (is a group), the complexity is trivially $O(n)$

Problem: Suppose $A \in \{0,1\}^{n \times n}$ has z zeros. How many operations do we need to compute Ax as a function of z?

Motivation

- Want to understand the structure of semigroups that are not groups

Motivation

- Want to understand the structure of semigroups that are not groups
- Important examples: Boolean semigroup and tropical semigroup

Motivation

- Want to understand the structure of semigroups that are not groups
- Important examples: Boolean semigroup and tropical semigroup
- Famous problem of similar flavor: matrix multiplication
  Given two matrices A, B, how many operations are needed to compute $A \cdot B$?

Motivation

- Want to understand the structure of semigroups that are not groups
- Important examples: Boolean semigroup and tropical semigroup
- Famous problem of similar flavor: matrix multiplication
  Given two matrices A, B, how many operations are needed to compute $A \cdot B$?
- Non-trivial upper bound over integers: $O(n^{2.373})$ (V. Williams '12)

Motivation

- Want to understand the structure of semigroups that are not groups
- Important examples: Boolean semigroup and tropical semigroup
- Famous problem of similar flavor: matrix multiplication
  Given two matrices A, B, how many operations are needed to compute $A \cdot B$?
- Non-trivial upper bound over integers: $O(n^{2.373})$ (V. Williams '12)
- Known lower bound over tropical semiring: $\Omega(n^3)$ (Kerr '70)

Motivation

- Want to understand the structure of semigroups that are not groups
- Important examples: Boolean semigroup and tropical semigroup
- Famous problem of similar flavor: matrix multiplication
  Given two matrices A, B, how many operations are needed to compute $A \cdot B$?
- Non-trivial upper bound over integers: $O(n^{2.373})$ (V. Williams '12)
- Known lower bound over tropical semiring: $\Omega(n^3)$ (Kerr '70)
- Other motivation: connection to range minimum query problem (will see later)

Main results

Theorem

If $(S, \circ)$ is a commutative semigroup, then Ax can be computed in $O(n)$ operations for dense A

Main results

Theorem

If $(S, \circ)$ *is a commutative semigroup, then* Ax *can be computed in* $O(n)$ *operations for dense* A

More generally,

Theorem

If $(S, \circ)$ *is a commutative semigroup, and* A *has* z *zeros, then* Ax *can be computed in* $O(z)$ *operations*

Main results

Theorem

If $(S, \circ)$ *is a commutative semigroup, then* Ax *can be computed in* $O(n)$ *operations for dense* A

More generally,

Theorem

If $(S, \circ)$ *is a commutative semigroup, and* A *has* z *zeros, then* Ax *can be computed in* $O(z)$ *operations*

Theorem

If $(S, \circ)$ *is strongly non-commutative semigroup, then the maximal complexity of* Ax *is* $\Theta(n\alpha(n))$ *operations for dense* A

Here $\alpha(n)$ is the inverse Ackermann function

Main results

Theorem
If $(S, \circ)$ is a commutative semigroup, then Ax can be computed in $O(n)$ operations for dense A

More generally,

Theorem
If $(S, \circ)$ is a commutative semigroup, and A has z zeros, then Ax can be computed in $O(z)$ operations

Theorem
If $(S, \circ)$ is strongly non-commutative semigroup, then the maximal complexity of Ax is $\Theta(n\alpha(n))$ operations for dense A

Here $\alpha(n)$ is the inverse Ackermann function

$\alpha(n)$ growth is extremely slow

For all practical needs we can assume $\alpha(n) \leqslant 4$

Connection to RMQ

Theorem (simplified upper bound)

For dense A the operator Ax over $(\{0,1\}, \vee)$ can be computed is $O(n)$ operations

First idea: Connection to Range Minimum Query problem (RMQ)

This is a standard setting in theory of algorithms

We are given an array of numbers $x_1, \ldots, x_n$. We want a data structure to answer queries of the form

$$\min\{x_i \mid l \leqslant i \leqslant r\} = ?$$

for integer l and r.

Reduction to RMQ

Consider

$$A = \begin{bmatrix} 1 & 0 & 1 & 1 & 0 & 1 \\ 1 & 1 & 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 & 1 & 1 \end{bmatrix}$$

Split each row into intervals

$$\begin{array}{lll} x_1 & x_3 \vee x_4 & x_6 \\ x_1 \vee x_2 & x_4 \vee x_5 \vee x_6 & \\ x_1 & x_3 \vee x_4 \vee x_5 & \\ x_1 \vee x_2 \vee x_3 & x_5 \vee x_6 & \\ x_2 \vee x_3 \vee x_4 \vee x_5 \vee x_6 & & \\ x_1 & x_4 \vee x_5 \vee x_6 & \end{array}$$

There are $O(n)$ intervals in total, so we reduced our problem to the offline version of RMQ (intervals are given in advance)

Complexity of RMQ

Unfortunately, best constructions for RMQ give only $O(n\alpha(n))$ complexity in our model, where $\alpha(n)$ is an inverse Ackermann function

Complexity of RMQ

Unfortunately, best constructions for RMQ give only $O(n\alpha(n))$ complexity in our model, where $\alpha(n)$ is an inverse Ackermann function

Moreover, the following is true

Theorem (Chazelle, Rozenberg '91)

There are range matrices $A \in \{0,1\}^{n \times n}$ with the complexity $\Omega(n\alpha(n))$

So, the reduction to RMQ is not enough for the upper bound

Upper Bound Proof Idea

$$A = \begin{bmatrix} * & * & * & * & * & * \\ * & * & * & * & * & * \\ * & * & * & * & * & * \\ * & * & * & * & * & * \\ * & * & * & * & * & * \\ * & * & * & * & * & * \end{bmatrix}$$

Suppose we have A with $O(n)$ zeros

Upper Bound Proof Idea

$$A = \left[\begin{array}{cccccc} * & * & * & * & * & * \\ * & * & * & * & * & * \\ \hline * & * & * & * & * & * \\ * & * & * & * & * & * \\ * & * & * & * & * & * \\ * & * & * & * & * & * \end{array}\right]$$

Suppose we have A with $O(n)$ zeros

- Split rows in two parts: with more than $\log n$ zeros and with at most $\log n$ zeros

Upper Bound Proof Idea

$$A = \begin{bmatrix} * & * & * & * & * & * \\ * & * & * & * & * & * \\ \hline * & * & * & * & * & * \\ * & * & * & * & * & * \\ * & * & * & * & * & * \\ * & * & * & * & * & * \end{bmatrix}$$

Suppose we have A with $O(n)$ zeros

- Split rows in two parts: with more than $\log n$ zeros and with at most $\log n$ zeros
- Computing the second part: intervals are long on average

Upper Bound Proof Idea

$$A = \left[\begin{array}{cccccc} * & * & * & * & * & * \\ * & * & * & * & * & * \\ \hline * & * & * & * & * & * \\ * & * & * & * & * & * \\ * & * & * & * & * & * \\ * & * & * & * & * & * \end{array}\right]$$

Suppose we have A with $O(n)$ zeros

- Split rows in two parts: with more than $\log n$ zeros and with at most $\log n$ zeros
- Computing the second part: intervals are long on average
- The first part has at most $n/\log n$ rows

Upper Bound Proof Idea

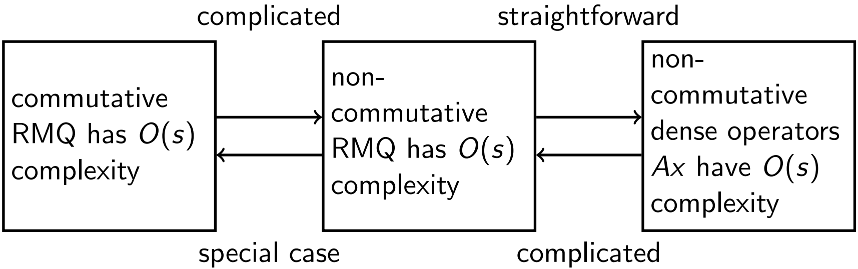
$$A = \left[\begin{array}{cccccc} * & * & * & * & * & * \\ * & * & * & * & * & * \\ \hline * & * & * & * & * & * \\ * & * & * & * & * & * \\ * & * & * & * & * & * \\ * & * & * & * & * & * \end{array}\right]$$

Suppose we have A with $O(n)$ zeros

- Split rows in two parts: with more than $\log n$ zeros and with at most $\log n$ zeros
- Computing the second part: intervals are long on average
- The first part has at most $n/\log n$ rows
- Computing the first part: switch to the transposed matrix

Lower Bound, Proof Scheme

We prove the following problem equivalences



Open problem

Problem (Jukna '19)

Consider $A \in \{0,1\}^{n\times n}$, *denote by* $\overline{A}$ *the bit-wise negation of* A. *How large can*

$$\frac{\mathit{Complexity}(\overline{A}x)}{\mathit{Complexity}(Ax)}$$

be over $(\mathbb{N},+)$ *semiring?*

Open problem

Problem (Jukna '19)

Consider $A \in \{0, 1\}^{n \times n}$*, denote by* $\overline{A}$ *the bit-wise negation of* A*. How large can*

$$\frac{\mathit{Complexity}(\overline{A}x)}{\mathit{Complexity}(Ax)}$$

be over $(\mathbb{N}, +)$ *semiring?*

Our result rules out the possibility to achieve large gap with a sparse matrix

Open problem

Problem (Jukna '19)

Consider $A \in \{0,1\}^{n\times n}$*, denote by* $\overline{A}$ *the bit-wise negation of* A*. How large can*

$$\frac{Complexity(\overline{A}x)}{Complexity(Ax)}$$

be over $(\mathbb{N}, +)$ *semiring?*

Our result rules out the possibility to achieve large gap with a sparse matrix

Thank you for attention!