A study of the rank computation problem for linear matrices

Somnath Bhattacharjee
(Chennai Mathematical Institute)

under supervision of
Prof. Partha Mukhopadhyay

May 30, 2024

Outline

Commutative Rank

Edmonds' Problem

Given a polynomial matrix $A(X_1, \ldots, X_n)$ with linear entries

$$A(\mathbf{X}) = A_1 X_1 + A_2 X_2 + \cdots + A_n X_n \quad A_i \in \mathbb{C}^{s \times s}$$

Find the rank of $A(\mathbf{X})$ over $\mathbb{C}(\mathbf{X})$. Input parameter: n, s

[1] can be solved using PIT oracle

Commutative Rank

Edmonds' Problem

Given a polynomial matrix $A(X_1, \dots, X_n)$ with linear entries

$$A(\mathbf{X}) = A_1 X_1 + A_2 X_2 + \cdots + A_n X_n \quad A_i \in \mathbb{C}^{s \times s}$$

Find the rank of $A(\mathbf{X})$ over $\mathbb{C}(\mathbf{X})$. Input parameter: n, s

- Has a randomised algorithm[1]

[1] can be solved using PIT oracle

Commutative Rank

Edmonds' Problem

Given a polynomial matrix $A(X_1, \ldots, X_n)$ with linear entries

$$A(\mathbf{X}) = A_1X_1 + A_2X_2 + \cdots + A_nX_n \quad A_i \in \mathbb{C}^{s\times s}$$

Find the rank of $A(\mathbf{X})$ over $\mathbb{C}(\mathbf{X})$. Input parameter: n, s

- Has a randomised algorithm[1]
- Polytime algorithm known for some restrictive cases

[1] can be solved using PIT oracle

Commutative Rank

Edmonds' Problem

Given a polynomial matrix $A(X_1, \ldots, X_n)$ with linear entries

$$A(\mathbf{X}) = A_1X_1 + A_2X_2 + \cdots + A_nX_n \quad A_i \in \mathbb{C}^{s \times s}$$

Find the rank of $A(\mathbf{X})$ over $\mathbb{C}(\mathbf{X})$. Input parameter: n, s

- Has a randomised algorithm[1]
- Polytime algorithm known for some restrictive cases
- Deterministic Polytime Approximation Scheme (PTAS) known due to Bläser, Jindal, Pandey (2016)

[1] can be solved using PIT oracle

PTAS

ε - Approximation

Given a constant $0 < \varepsilon < 1$ and input $A(\mathbf{X})$ we can output a number r s.t.

$$r \leq \mathrm{crk}(A) \leq r[1 + \varepsilon]$$

in poly $((ns)^{1/\varepsilon})$ time

Non-commutative Rank

Non-commutative Edmonds' Problem

Given a polynomial matrix $A(X_1, \ldots, X_n)$ with linear entries

$$A(\mathbf{X}) = A_1 X_1 + A_2 X_2 + \cdots + A_n X_n \quad A_i \in \mathbb{C}^{s \times s}$$

Non-commutative Rank

Non-commutative Edmonds' Problem

Given a polynomial matrix $A(X_1, \ldots, X_n)$ with linear entries

$$A(\mathbf{X}) = A_1X_1 + A_2X_2 + \cdots + A_nX_n \quad A_i \in \mathbb{C}^{s\times s}$$

Where **X** are **non-commutative**

Non-commutative Rank

Non-commutative Edmonds' Problem

Given a polynomial matrix $A(X_1, \ldots, X_n)$ with linear entries

$$A(\mathbf{X}) = A_1 X_1 + A_2 X_2 + \cdots + A_n X_n \quad A_i \in \mathbb{C}^{s \times s}$$

Where $\mathbf{X}$ are **non-commutative**

Find the rank of $A(\mathbf{X})$ over $\mathbb{C}\langle(\mathbf{X})\rangle$ (**non-commutative rank**).

Non-commutative Rank

Non-commutative rank of a matrix in $\mathbb{C}\langle(\mathbf{X})\rangle$ is

Row rk : Max linear independent rows under left action

Non-commutative Rank

Non-commutative rank of a matrix in $\mathbb{C}\langle(\mathbf{X})\rangle$ is

Row rk : Max linear independent rows under left action

Column rk :

Non-commutative Rank

Non-commutative rank of a matrix in $\mathbb{C}\langle(\mathbf{X})\rangle$ is

Row rk : Max linear independent rows under left action

Column rk :

Inner rk : min r s.t. A can be written as product of $n \times r$ and $r \times n$ matrix

Non-commutative Rank

Non-commutative rank of a matrix in $\mathbb{C}\langle(\mathbf{X})\rangle$ is

Row rk : Max linear independent rows under left action

Column rk :

Inner rk : min r s.t. A can be written as product of $n \times r$ and $r \times n$ matrix

- Max r s.t. it has $r \times r$ full non-commutative rank minor

Non-commutative Rank

Example:

$$\begin{bmatrix} x & 0 & 0 \\ 0 & y & 0 \\ 0 & 0 & z \end{bmatrix}$$

Non-commutative Rank

Example:

$$\begin{bmatrix} x & 0 & 0 \\ 0 & y & 0 \\ 0 & 0 & z \end{bmatrix}$$

Suppose

$$\begin{bmatrix} x \\ 0 \\ 0 \end{bmatrix} a + \begin{bmatrix} 0 \\ y \\ 0 \end{bmatrix} b + \begin{bmatrix} 0 \\ 0 \\ z \end{bmatrix} c = 0$$

Non-commutative Rank

Example:

$$\begin{bmatrix} x & 0 & 0 \\ 0 & y & 0 \\ 0 & 0 & z \end{bmatrix}$$

Suppose

$$\begin{bmatrix} x \\ 0 \\ 0 \end{bmatrix} a + \begin{bmatrix} 0 \\ y \\ 0 \end{bmatrix} b + \begin{bmatrix} 0 \\ 0 \\ z \end{bmatrix} c = 0$$

$$\Longrightarrow \begin{bmatrix} xa \\ yb \\ zc \end{bmatrix} = 0 \quad \Longrightarrow a = b = c = 0$$

Hence has non-commutative rank 3

Non-commutative Rank

Example:

$$\begin{bmatrix} 0 & x & y \\ -x & 0 & 1 \\ -y & -1 & 0 \end{bmatrix}$$

Non-commutative Rank

Example:

$$\begin{bmatrix} 0 & x & y \\ -x & 0 & 1 \\ -y & -1 & 0 \end{bmatrix} \xrightarrow{c_3 \to c_3 - c_2 x^{-1} y} \begin{bmatrix} 0 & x & 0 \\ -x & 0 & 1 \\ -y & -1 & x^{-1} y \end{bmatrix}$$

Non-commutative Rank

Example:

$$\begin{bmatrix} 0 & x & y \\ -x & 0 & 1 \\ -y & -1 & 0 \end{bmatrix} \xrightarrow{c_3 \to c_3 - c_2 x^{-1} y} \begin{bmatrix} 0 & x & 0 \\ -x & 0 & 1 \\ -y & -1 & x^{-1}y \end{bmatrix}$$

$$\xrightarrow{c_1 \to c_1 - c_3 x} \begin{bmatrix} 0 & x & 0 \\ 0 & 0 & 1 \\ x^{-1}yx - y & -1 & x^{-1}y \end{bmatrix}$$

Non-commutative Rank

Example:

$$\begin{bmatrix} 0 & x & y \\ -x & 0 & 1 \\ -y & -1 & 0 \end{bmatrix} \xrightarrow{c_3 \to c_3 - c_2 x^{-1} y} \begin{bmatrix} 0 & x & 0 \\ -x & 0 & 1 \\ -y & -1 & x^{-1}y \end{bmatrix}$$

$$\xrightarrow{c_1 \to c_1 - c_3 x} \begin{bmatrix} 0 & x & 0 \\ 0 & 0 & 1 \\ x^{-1}yx - y & -1 & x^{-1}y \end{bmatrix} \xrightarrow{***} \begin{bmatrix} 0 & x & 0 \\ 0 & 0 & 1 \\ x^{-1}yx - y & 0 & 0 \end{bmatrix}$$

Hence has non-commutative rank 3

Non-commutative Rank

Example:

$$\begin{bmatrix} 0 & x & y \\ -x & 0 & 1 \\ -y & -1 & 0 \end{bmatrix} \xrightarrow{c_3 \to c_3 - c_2 x^{-1} y} \begin{bmatrix} 0 & x & 0 \\ -x & 0 & 1 \\ -y & -1 & x^{-1}y \end{bmatrix}$$

$$\xrightarrow{c_1 \to c_1 - c_3 x} \begin{bmatrix} 0 & x & 0 \\ 0 & 0 & 1 \\ x^{-1}yx - y & -1 & x^{-1}y \end{bmatrix} \xrightarrow{***} \begin{bmatrix} 0 & x & 0 \\ 0 & 0 & 1 \\ x^{-1}yx - y & 0 & 0 \end{bmatrix}$$

Hence has non-commutative rank 3

Note it's **commutative rank** is 2

Non-commutative Rank

Non-commutative Edmonds' problem has polytime algorithm due to

1. Garg, Gurvits, Oliveira and Wigderson 2015
2. Ivanyos, Qiao and Subrahmanyam 2015
3. Hamada and Hirai 2020

Partially Commutative Partition

$$\mathbf{X} := \{X_1, X_2, \ldots, X_m, X_{m+1}, \ldots, X_{2m}, \ldots, X_n\}$$

Partially Commutative Partition

$$\mathbf{X} := \{X_1, X_2, \ldots, X_m, X_{m+1}, \ldots, X_{2m}, \ldots, X_n\}$$

$$\underbrace{X_1, X_2, \ldots, X_m}_{\mathbf{X}_1} \Big| , \underbrace{X_{m+1}, \ldots, X_{2m}}_{\mathbf{X}_2} \Big| , \ldots, X_n$$

$$\mathbf{X} := \mathbf{X}_1 \sqcup \mathbf{X}_2 \sqcup \ldots \sqcup \mathbf{X}_k$$

Partially Commutative Partition

$$\mathbf{X} := \{X_1, X_2, \ldots, X_m, X_{m+1}, \ldots, X_{2m}, \ldots, X_n\}$$

$$\underbrace{X_1, X_2, \ldots, X_m}_{\mathbf{X}_1} \Big| , \underbrace{X_{m+1}, \ldots, X_{2m}}_{\mathbf{X}_2} \Big| , \ldots, X_n$$

$$\mathbf{X} := \mathbf{X}_1 \sqcup \mathbf{X}_2 \sqcup \ldots \sqcup \mathbf{X}_k$$

$$\mathbf{X}_i \text{ commutes with } \mathbf{X}_j \iff i \neq j$$

Partially Commutative Partition

$$\mathbf{X} := \{X_1, X_2, \ldots, X_m, X_{m+1}, \ldots, X_{2m}, \ldots, X_n\}$$

$$\underbrace{X_1, X_2, \ldots, X_m}_{\mathbf{X}_1} \Big| , \underbrace{X_{m+1}, \ldots, X_{2m}}_{\mathbf{X}_2} \Big| , \ldots, X_n$$

$$\mathbf{X} := \mathbf{X}_1 \sqcup \mathbf{X}_2 \sqcup \ldots \sqcup \mathbf{X}_k$$

$$\mathbf{X}_i \text{ commutes with } \mathbf{X}_j \iff i \neq j$$

Call this a **k - Partially Commutative Partition**

Partially Commutative Partition

- For any k - Partially Commutative Partition, we have a universal skew field, denoted by $\mathfrak{U}_k$ (Klep et all. 2020)

Partially Commutative Partition

- For any k - Partially Commutative Partition, we have a universal skew field, denoted by $\mathfrak{U}_k$ (Klep et all. 2020)
- Hence we can define rank problem in this model

Partially Commutative Partition

- For any k - Partially Commutative Partition, we have a universal skew field, denoted by $\mathfrak{U}_k$ (Klep et all. 2020)
- Hence we can define rank problem in this model
- Arvind, Chatterjee and Mukhopadhyay (2024) gave a $O((ns)^{k^k})$ algorithm.

Partially Commutative Partition

- For any k - Partially Commutative Partition, we have a universal skew field, denoted by $\mathfrak{U}_k$ (Klep et all. 2020)
- Hence we can define rank problem in this model
- Arvind, Chatterjee and Mukhopadhyay (2024) gave a $O((ns)^{k^k})$ algorithm.
- We can design $O(poly(nsk))$ time approximation algorithm.

Brief Timeline

IQS 15 : polynomial time algorithm for Non-commutative rank

Brief Timeline

IQS 15 : polynomial time algorithm for Non-commutative rank

BJP 16 : PTAS for commutative rank

Brief Timeline

IQS 15 : polynomial time algorithm for Non-commutative rank

BJP 16 : PTAS for commutative rank

BBJP 18 : Simplified [BJP 18]

Brief Timeline

IQS 15 : polynomial time algorithm for Non-commutative rank

BJP 16 : PTAS for commutative rank

BBJP 18 : Simplified [BJP 18]

CM 23 : Simplified [IQS 15]

Brief Timeline

IQS 15 : polynomial time algorithm for Non-commutative rank

BJP 16 : PTAS for commutative rank

BBJP 18 : Simplified [BJP 18]

CM 23 : Simplified [IQS 15]

- Based on the ideas from last two results: $poly(kns)$ approximation algorithm for Partially Commutative model

ABP

Algebraic Branching Program

Product of $t+2$ many $s \times s$ Matrix polynomials with linear entries:

$$\begin{bmatrix} r_1 & \dots & r_s \end{bmatrix} \begin{bmatrix} l_{11}^{(1)} & \dots & l_{1s}{}^{(1)} \\ \vdots & & \\ l_{s1}{}^{(1)} & \dots & l_{ss}{}^{(1)} \end{bmatrix} \dots \begin{bmatrix} l_{11}^{(t)} & \dots & l_{1s}{}^{(t)} \\ \vdots & & \\ l_{s1}{}^{(t)} & \dots & l_{ss}{}^{(t)} \end{bmatrix} \begin{bmatrix} c_1 \\ \vdots \\ c_s \end{bmatrix}$$

$r_i, l_{jk}^{(i)}, c_i$ are linear polynomials

ABP

Algebraic Branching Program

Product of $t+2$ many $s \times s$ Matrix polynomials with linear entries:

$$\begin{bmatrix} r_1 & \dots & r_s \end{bmatrix} \begin{bmatrix} l_{11}^{(1)} & \dots & {l_{1s}}^{(1)} \\ \vdots & & \\ {l_{s1}}^{(1)} & \dots & {l_{ss}}^{(1)} \end{bmatrix} \cdots \begin{bmatrix} l_{11}^{(t)} & \dots & {l_{1s}}^{(t)} \\ \vdots & & \\ {l_{s1}}^{(t)} & \dots & {l_{ss}}^{(t)} \end{bmatrix} \begin{bmatrix} c_1 \\ \vdots \\ c_s \end{bmatrix}$$

$r_i, l_{jk}^{(i)}, c_i$ are linear polynomials

- The polynomial computed by the ABP is the product polynomial

ABP

Algebraic Branching Program

Product of $t+2$ many $s \times s$ Matrix polynomials with linear entries:

$$\begin{bmatrix} r_1 & \dots & r_s \end{bmatrix} \begin{bmatrix} l_{11}^{(1)} & \dots & {l_{1s}}^{(1)} \\ \vdots & & \\ {l_{s1}}^{(1)} & \dots & {l_{ss}}^{(1)} \end{bmatrix} \cdots \begin{bmatrix} l_{11}^{(t)} & \dots & {l_{1s}}^{(t)} \\ \vdots & & \\ {l_{s1}}^{(t)} & \dots & {l_{ss}}^{(t)} \end{bmatrix} \begin{bmatrix} c_1 \\ \vdots \\ c_s \end{bmatrix}$$

$r_i, l_{jk}^{(i)}, c_i$ are linear polynomials

- The polynomial computed by the ABP is the product polynomial
- r is the length and s is the width of the ABP

Polynomial Identity Testing

PIT for ABP

Given an ABP of length l, width s, check whether the polynomial is 0 or not in $poly(l, s, n)$ time

Polynomial Identity Testing

PIT for ABP

Given an ABP of length l, width s, check whether the polynomial is 0 or not in $poly(l, s, n)$ time

- if l is constant we can do the PIT efficiently

Polynomial Identity Testing

PIT for ABP

Given an ABP of length l, width s, check whether the polynomial is 0 or not in $poly(l, s, n)$ time

- if l is constant we can do the PIT efficiently
- Efficient PIT known for Non-commutative ABP due to Raz and Shpilka (2004)

Non commutative rank vs Commutative rank

Theorem 1

$\text{crk}(A(\mathbf{X})) := \text{max rank in}\langle A_1, A_2, \ldots, A_n \rangle$

Non commutative rank vs Commutative rank

Theorem 1

$\text{crk}(A(\mathbf{X})) := \text{max rank in} \langle A_1, A_2, \ldots, A_n \rangle$

Which is the max rank obtained after substituting *scaler values* for **X**

Non commutative rank vs Commutative rank

Theorem 1

$\text{crk}(A(\mathbf{X})) := \text{max rank in} \langle A_1, A_2, \ldots, A_n \rangle$

Which is the max rank obtained after substituting *scaler values* for $\mathbf{X}$

$$\implies \text{crk}(A(\mathbf{X})) = \text{crk}(A(\mathbf{X} + \alpha))$$

Non commutative rank vs Commutative rank

Theorem 1

$\text{crk}(A(\mathbf{X})) := \text{max rank in}\langle A_1, A_2, \ldots, A_n \rangle$

Which is the max rank obtained after substituting *scaler values* for **X**

$$\implies \text{crk}(A(\mathbf{X})) = \text{crk}(A(\mathbf{X} + \alpha))$$

Theorem 2 (informal)

ncrk(A) is max rank obtained when we substitute matrices for **X** (and tensoring with A_is)

Non commutative rank vs Commutative rank

Theorem 3

$$\mathrm{crk}(A) \leq \mathrm{ncrk}(A) \leq 2\mathrm{crk}(A)$$

Non commutative rank vs Commutative rank

Theorem 3 (General statement)

For $k \leq t$, if t - Partially commutative partition refines k - Partially commutative partition

Non commutative rank vs Commutative rank

Theorem 3 (General statement)

For $k \leq t$, if t - Partially commutative partition refines k - Partially commutative partition

$$\text{rk}_t(A) \leq \text{rk}_k(A) \leq 2\text{rk}_t(A)$$

Non commutative rank vs Commutative rank

Theorem 3 (General statement)

For $k \leq t$, if t - Partially commutative partition refines k - Partially commutative partition

$$\mathrm{rk}_t(A) \leq \mathrm{rk}_k(A) \leq 2\mathrm{rk}_t(A)$$

Note it already gives an approximation algorithm with fixed error

Non commutative rank vs Commutative rank

Theorem 3 (General statement)

For $k \leq t$, if t - Partially commutative partition refines k - Partially commutative partition

$$\text{rk}_t(A) \leq \text{rk}_k(A) \leq 2\text{rk}_t(A)$$

Note it already gives an approximation algorithm with fixed error
- Output ncrk and we know it is close to rk_k

Basic structure of the Algorithm

The Algorithm will be greedy:

- Base case: Start with a 1×1 minor

Basic structure of the Algorithm

The Algorithm will be greedy:

- Base case: Start with a 1×1 minor
- Rank increment: given an assignment $\alpha \in \mathbb{C}^n$ s.t. $\text{crk}(A(\alpha)) = r$, find β greedily s.t. $\text{crk}(A(\beta)) > r$

Basic structure of the Algorithm

The Algorithm will be greedy:

- Base case: Start with a 1×1 minor
- Rank increment: given an assignment $\alpha \in \mathbb{C}^n$ s.t. $\mathrm{crk}(A(\alpha)) = r$, find β greedily s.t. $\mathrm{crk}(A(\beta)) > r$
- Rank approximation: If no such β, output r

Rank increment step

Given an assignment α for witness of rank r and ε, construct polynomially many ABPs $\{f_{ij}\}$ s.t. :

Rank increment step

Given an assignment α for witness of rank r and ε, construct polynomially many ABPs $\{f_{ij}\}$ s.t. :

- f_{ij} has efficient PIT

Rank increment step

Given an assignment α for witness of rank r and ε, construct polynomially many ABPs $\{f_{ij}\}$ s.t. :

- f_{ij} has efficient PIT
- If one of the $f_{ij} \neq 0$, we can find β s.t. $\text{crk}(A(\beta)) > r$

Rank increment step

Given an assignment α for witness of rank r and ε, construct polynomially many ABPs $\{f_{ij}\}$ s.t. :

- f_{ij} has efficient PIT
- If one of the $f_{ij} \neq 0$, we can find β s.t. $\mathrm{crk}(A(\beta)) > r$
- Else $\mathrm{crk}(A) \leq r(1+\varepsilon)$

Rank increment step - Analysis

WLOG assume

$$A(\alpha) = \begin{bmatrix} I_r & 0 \\ 0 & 0 \end{bmatrix}$$

Rank increment step - Analysis

WLOG assume

$$A(\alpha) = \begin{bmatrix} I_r & 0 \\ 0 & 0 \end{bmatrix}$$

Hence

$$A(\mathbf{X} + \alpha) = \begin{matrix} r \text{ rows} \\ n - r \text{ rows} \end{matrix} \begin{matrix} \{ \\ \{ \end{matrix} \underbrace{\begin{bmatrix} I_r - L(\mathbf{X}) & B(\mathbf{X}) \\ D(\mathbf{X}) & C(\mathbf{X}) \end{bmatrix}}_{r \qquad \text{n-r}}$$

Where $L(\mathbf{X}), B(\mathbf{X}), C(\mathbf{X})$ are matrix polynomials with constant free linear entries.

Rank increment step - Analysis

Suppose $\exists i, j \in [n-r]$ s.t.

$$\begin{vmatrix} I_r - L(\mathbf{X}) & B_j(\mathbf{X}) \\ D_i(\mathbf{X}) & C_{ij}(\mathbf{X}) \end{vmatrix} \neq 0$$

Rank increment step - Analysis

Suppose $\exists i, j \in [n-r]$ s.t.

$$\begin{vmatrix} I_r - L(\mathbf{X}) & B_j(\mathbf{X}) \\ D_i(\mathbf{X}) & C_{ij}(\mathbf{X}) \end{vmatrix} \neq 0$$

$$\iff C_{ij}(\mathbf{X}) - D_i(\mathbf{X})[I - L(\mathbf{X})]^{-1} B_j(\mathbf{X}) \neq 0$$

Rank increment step - Analysis

Suppose $\exists i, j \in [n-r]$ s.t.

$$\begin{vmatrix} I_r - L(\mathbf{X}) & B_j(\mathbf{X}) \\ D_i(\mathbf{X}) & C_{ij}(\mathbf{X}) \end{vmatrix} \neq 0$$

$$\iff C_{ij}(\mathbf{X}) - D_i(\mathbf{X})[I - L(\mathbf{X})]^{-1} B_j(\mathbf{X}) \neq 0$$

$$\iff C_{ij} - D_i(\sum_{t \geq 0} L^t) B_j) \neq 0$$

Rank increment step - Analysis

Suppose $\exists i, j \in [n-r]$ s.t.

$$\begin{vmatrix} I_r - L(\mathbf{X}) & B_j(\mathbf{X}) \\ D_i(\mathbf{X}) & C_{ij}(\mathbf{X}) \end{vmatrix} \neq 0$$

$$\iff C_{ij}(\mathbf{X}) - D_i(\mathbf{X})[I - L(\mathbf{X})]^{-1} B_j(\mathbf{X}) \neq 0$$

$$\iff C_{ij} - D_i(\sum_{t \geq 0} L^t) B_j) \neq 0$$

$$\iff C_{ij} - D_i(\sum_{t=0}^{\mathbf{s}} L^t) B_j \neq 0$$

Rank increment step - Analysis

$$\mathrm{crk}(A) = r \iff C - \sum_{t=0}^{s} BL^t D = 0$$

Rank increment step - Analysis

$$\mathrm{crk}(A) = r \iff C - \sum_{t=0}^{s} BL^t D = 0$$

$$\iff C, BD, BLD, \ldots, BL^s D = 0$$

Rank increment step - Analysis

$$\mathrm{crk}(A) = r \iff C - \sum_{t=0}^{s} BL^t D = 0$$

$$\iff C, BD, BLD, \ldots, BL^s D = 0$$

Bläser, Jindal, Pandey 2016:

$$C, BD, \ldots, BL^{k-2} D = 0 \implies \mathrm{crk}(A) \leq r(1 + \frac{1}{k})$$

Rank increment step - Analysis

$$\text{crk}(A) = r \iff C - \sum_{t=0}^{s} BL^t D = 0$$

$$\iff C, BD, BLD, \ldots, BL^s D = 0$$

Bläser, Jindal, Pandey 2016:

$$C, BD, \ldots, BL^{k-2} D = 0 \implies \text{crk}(A) \leq r(1 + \frac{1}{k})$$

Final Algorithm:

- Set $k = \frac{1}{\varepsilon}$
- Check for $C, BD, \ldots, BL^{k-2} D = 0$. Output accordingly

Rank increment step - Analysis

Suppose some $BL^tD \neq 0$ for $\mathbf{X} = \gamma$,

Rank increment step - Analysis

Suppose some $BL^tD \neq 0$ for $\mathbf{X} = \gamma$,

- That means $A(\alpha + t\gamma)$ has non-zero $r + 1 \times r + 1$ minor

Rank increment step - Analysis

Suppose some $BL^tD \neq 0$ for $\mathbf{X} = \gamma$,

- That means $A(\alpha + t\gamma)$ has non-zero $r+1 \times r+1$ minor
- The determinant of that minor is $r+1$ degree univariate polynomial

Rank increment step - Analysis

Suppose some $BL^tD \neq 0$ for $\mathbf{X} = \gamma$,

- That means $A(\alpha + t\gamma)$ has non-zero $r + 1 \times r + 1$ minor
- The determinant of that minor is $r + 1$ degree univariate polynomial
- Assigning different $r + 2$ many values to t will assure one of them makes the minor non-zero.

High level idea for non-commutative case

For $d \geq s+1$, let Z_i be $d \times d$ variable matrices

$$\tilde{A}_d(\mathbf{Z}_1, \ldots, \mathbf{Z}_n)_{ds \times ds} := A_1 \otimes \mathbf{Z}_1 + \cdots + A_n \otimes \mathbf{Z}_n$$

High level idea for non-commutative case

For $d \geq s+1$, let Z_i be $d \times d$ variable matrices

$$\tilde{A}_d(\mathbf{Z}_1, \ldots, \mathbf{Z}_n)_{ds \times ds} := A_1 \otimes \mathbf{Z}_1 + \cdots + A_n \otimes \mathbf{Z}_n$$

Theorem 2 (formal)

$$\text{ncrk}(A(\mathbf{X})) = d \times \text{ncrk}(\tilde{A}_d(\mathbf{Z}))$$

High level idea for non-commutative case

Let $(\alpha_1, \ldots, \alpha_n) \in [\mathbb{C}^{d\times d}]^n$ be a witness assignment for rank r

High level idea for non-commutative case

Let $(\alpha_1, \ldots, \alpha_n) \in [\mathbb{C}^{d\times d}]^n$ be a witness assignment for rank r

$$\tilde{A}_d(\mathbf{Z} + \alpha) = \begin{bmatrix} I_{rd} - L & B \\ D & C \end{bmatrix}_{sd\times sd}$$

High level idea for non-commutative case

Let $(\alpha_1, \ldots, \alpha_n) \in [\mathbb{C}^{d\times d}]^n$ be a witness assignment for rank r

$$\tilde{A}_d(\mathbf{Z} + \alpha) = \begin{bmatrix} I_{rd} - L & B \\ D & C \end{bmatrix}_{sd\times sd}$$

- Hence we have the series $C, BD, \ldots, BL^{sd}D$

High level idea for non-commutative case

Let $(\alpha_1, \ldots, \alpha_n) \in [\mathbb{C}^{d \times d}]^n$ be a witness assignment for rank r

$$\tilde{A}_d(\mathbf{Z} + \alpha) = \begin{bmatrix} I_{rd} - L & B \\ D & C \end{bmatrix}_{sd \times sd}$$

- Hence we have the series $C, BD, \ldots, BL^{sd} D$
- PIT is easy

High level idea for non-commutative case

Let $(\alpha_1, \ldots, \alpha_n) \in [\mathbb{C}^{d \times d}]^n$ be a witness assignment for rank r

$$\tilde{A}_d(\mathbf{Z} + \alpha) = \begin{bmatrix} I_{rd} - L & B \\ D & C \end{bmatrix}_{sd \times sd}$$

- Hence we have the series $C, BD, \ldots, BL^{sd}D$
- PIT is easy
- If first k many terms are 0

$$rd \leq \text{ncrk}(\tilde{A}_d) \leq rd(1 + \frac{1}{k})$$

$$\implies r \leq \text{ncrk}(A) \leq r(1 + \frac{1}{k})$$

Substitution in Partially Commutative model

Given

$$\mathbf{X} = \mathbf{X}_1 \sqcup \cdots \sqcup \mathbf{X}_k$$

define

$$\tilde{A}(\mathbf{Z}) := \sum_{i=1}^{k} \sum_{x \in \mathbf{X}_i} A_X \otimes I_{d_1} \otimes \cdots \otimes I_{d_{i-1}} \otimes Z_X \otimes I_{d_{i+1}} \otimes \cdots \otimes I_{d_k}$$

Open areas

1. FPT algorithm for the rank problem in Partially commutative model
2. Hardness in partially commutative model. Does that generalise the extreme cases?
3. Efficiency for PC-rank computation for non-linear matrices