Matrix factorization ranks: why do we want to lower bound them?

Sander Gribling, CWI

Centrum Wiskunde & Informatica

Matrix factorization ranks

Given $A \in \mathbb{R}_{+}^{m \times n}$:

Matrix factorization ranks

Given $A \in \mathbb{R}_+^{m \times n}$:

$\mathrm{rank}(A) \quad = \min\{d : a_1, \ldots, a_m, b_1, \ldots, b_n \in \mathbb{R}^d \text{ with } A_{ij} = a_i^\mathsf{T} b_j\}$

Matrix factorization ranks

Given $A \in \mathbb{R}_+^{m \times n}$:

$$\begin{aligned}
\mathrm{rank}(A) \quad &= \min\{d : a_1, \ldots, a_m, b_1, \ldots, b_n \in \mathbb{R}^d \text{ with } A_{ij} = a_i^\mathsf{T} b_j\} \\
\mathrm{rank}_+(A) &= \min\{d : a_1, \ldots, a_m, b_1, \ldots, b_n \in \mathbb{R}_+^d \text{ with } A_{ij} = a_i^\mathsf{T} b_j\}
\end{aligned}$$

Matrix factorization ranks

Given $A \in \mathbb{R}_+^{m \times n}$:

$\mathrm{rank}(A) \quad = \min\{d : a_1, \ldots, a_m, b_1, \ldots, b_n \in \mathbb{R}^d \text{ with } A_{ij} = a_i^\mathsf{T} b_j\}$

$\mathrm{rank}_+(A) = \min\{d : a_1, \ldots, a_m, b_1, \ldots, b_n \in \mathbb{R}_+^d \text{ with } A_{ij} = a_i^\mathsf{T} b_j\}$

$A_{ij} = \mathrm{Tr}\Big(\mathrm{diag}(a_i)\mathrm{diag}(b_j)\Big)$

Matrix factorization ranks

Given $A \in \mathbb{R}_+^{m\times n}$:

$\text{rank}(A) \quad = \min\{d : a_1, \ldots, a_m, b_1, \ldots, b_n \in \mathbb{R}^d \text{ with } A_{ij} = a_i^\mathsf{T} b_j\}$

$\text{rank}_+(A) = \min\{d : a_1, \ldots, a_m, b_1, \ldots, b_n \in \mathbb{R}_+^d \text{ with } A_{ij} = a_i^\mathsf{T} b_j\}$

$$A_{ij} = \mathrm{Tr}\Big(\mathrm{diag}(a_i)\mathrm{diag}(b_j)\Big)$$

$\text{psd-rank}(A) = \min\{d : X_1, \ldots, X_m, Y_1, \ldots, Y_n \; d \times d \text{ Hermitian}$

$\text{PSD with } A_{ij} = \mathrm{Tr}(X_i Y_j)\}$

Matrix factorization ranks

Given $A \in \mathbb{R}_+^{m\times n}$:

$\mathrm{rank}(A) \quad = \min\{d : a_1, \ldots, a_m, b_1, \ldots, b_n \in \mathbb{R}^d \text{ with } A_{ij} = a_i^\mathsf{T} b_j\}$

$\mathrm{rank}_+(A) = \min\{d : a_1, \ldots, a_m, b_1, \ldots, b_n \in \mathbb{R}_+^d \text{ with } A_{ij} = a_i^\mathsf{T} b_j\}$

$A_{ij} = \mathrm{Tr}\Big(\mathrm{diag}(a_i)\mathrm{diag}(b_j)\Big)$

$\text{psd-rank}(A) = \min\{d : X_1, \ldots, X_m, Y_1, \ldots, Y_n \; d \times d \text{ Hermitian PSD with } A_{ij} = \mathrm{Tr}(X_i Y_j)\}$

Motivation:

Matrix factorization ranks

Given $A \in \mathbb{R}_+^{m\times n}$:

$$\mathrm{rank}(A) \quad = \min\{d : a_1,\ldots,a_m,b_1,\ldots,b_n \in \mathbb{R}^d \text{ with } A_{ij} = a_i^\mathsf{T} b_j\}$$

$$\mathrm{rank}_+(A) = \min\{d : a_1,\ldots,a_m,b_1,\ldots,b_n \in \mathbb{R}_+^d \text{ with } A_{ij} = a_i^\mathsf{T} b_j\}$$

$$A_{ij} = \mathrm{Tr}\Big(\mathrm{diag}(a_i)\mathrm{diag}(b_j)\Big)$$

$$\text{psd-rank}(A) = \min\{d : X_1,\ldots,X_m,Y_1,\ldots,Y_n \;\, d\times d \text{ Hermitian PSD with } A_{ij} = \mathrm{Tr}(X_i Y_j)\}$$

Motivation:

- linear/semidefinite extension complexity [Yannakakis'91,Gouveia-Parrilo-Thomas'13]

Matrix factorization ranks

Given $A \in \mathbb{R}_+^{m\times n}$:

$$\mathrm{rank}(A) \;\;= \min\{d : a_1,\ldots,a_m,b_1,\ldots,b_n \in \mathbb{R}^d \text{ with } A_{ij} = a_i^{\mathsf{T}} b_j\}$$

$$\mathrm{rank}_+(A) = \min\{d : a_1,\ldots,a_m,b_1,\ldots,b_n \in \mathbb{R}_+^d \text{ with } A_{ij} = a_i^{\mathsf{T}} b_j\}$$

$$A_{ij} = \mathrm{Tr}\Big(\mathrm{diag}(a_i)\mathrm{diag}(b_j)\Big)$$

$$\text{psd-rank}(A) = \min\{d : X_1,\ldots,X_m,Y_1,\ldots,Y_n \; d\times d \text{ Hermitian PSD with } A_{ij} = \mathrm{Tr}(X_i Y_j)\}$$

Motivation:

- linear/semidefinite extension complexity [Yannakakis'91,Gouveia-Parrilo-Thomas'13]
- (quantum) communication complexity [Fiorini-Massar-Pokutta-Tiwary-de Wolf'12]

Matrix factorization ranks

Given $A \in \mathbb{R}_+^{m \times n}$:

$\mathrm{rank}(A) \;\; = \min\{d : a_1, \ldots, a_m, b_1, \ldots, b_n \in \mathbb{R}^d \text{ with } A_{ij} = a_i^\mathsf{T} b_j\}$

$\mathrm{rank}_+(A) = \min\{d : a_1, \ldots, a_m, b_1, \ldots, b_n \in \mathbb{R}_+^d \text{ with } A_{ij} = a_i^\mathsf{T} b_j\}$

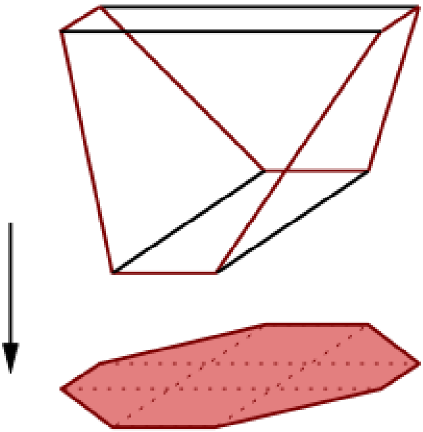
$A_{ij} = \mathrm{Tr}\Big(\mathrm{diag}(a_i)\mathrm{diag}(b_j)\Big)$

$\text{psd-rank}(A) = \min\{d : X_1, \ldots, X_m, Y_1, \ldots, Y_n \; d \times d \text{ Hermitian PSD with } A_{ij} = \mathrm{Tr}(X_i Y_j)\}$

Motivation:

- linear/semidefinite extension complexity [Yannakakis'91,Gouveia-Parrilo-Thomas'13]
- (quantum) communication complexity [Fiorini-Massar-Pokutta-Tiwary-de Wolf'12]
- clustering (rank_+ only)

Extension complexity



Extension complexity

$$B_1^n = \{x \in \mathbb{R}^n : \pm x_1 \pm \cdots \pm x_n \leq 1\}$$

Extension complexity

$$B_1^n = \{x \in \mathbb{R}^n : \underbrace{\pm x_1 \pm \cdots \pm x_n \leq 1}_{2^n \text{ inequalities}}\}$$

Extension complexity

$$B_1^n = \{x \in \mathbb{R}^n : \underbrace{\pm x_1 \pm \cdots \pm x_n \leq 1}_{2^n \text{ inequalities}}\}$$

$$Q_n = \{(x, y) \in \mathbb{R}^{2n} : \sum_{i=1}^{n} y_i = 1, \quad -y_i \leq x_i \leq y_i \text{ for } i \in [n]\}$$

Extension complexity

$$B_1^n = \{x \in \mathbb{R}^n : \underbrace{\pm x_1 \pm \cdots \pm x_n \le 1}_{2^n \text{ inequalities}}\}$$

$$Q_n = \{(x, y) \in \mathbb{R}^{2n} : \sum_{i=1}^{n} y_i = 1, \quad \underbrace{-y_i \le x_i \le y_i \text{ for } i \in [n]}_{2n \text{ inequalities}}\}$$

Extension complexity

$$B_1^n = \{x \in \mathbb{R}^n : \underbrace{\pm x_1 \pm \cdots \pm x_n \leq 1}_{2^n \text{ inequalities}}\}$$

$$Q_n = \{(x, y) \in \mathbb{R}^{2n} : \sum_{i=1}^{n} y_i = 1, \quad \underbrace{-y_i \leq x_i \leq y_i \text{ for } i \in [n]}_{2n \text{ inequalities}}\}$$

$\Rightarrow$ B_1^n is the projection of an affine slice of $\mathbb{R}_+^k$, where $k = 2n$.

Extension complexity

$$B_1^n = \{x \in \mathbb{R}^n : \underbrace{\pm x_1 \pm \cdots \pm x_n \leq 1}_{2^n \text{ inequalities}}\}$$

$$Q_n = \{(x, y) \in \mathbb{R}^{2n} : \sum_{i=1}^{n} y_i = 1, \quad \underbrace{-y_i \leq x_i \leq y_i \text{ for } i \in [n]}_{2n \text{ inequalities}}\}$$

$\Rightarrow$ B_1^n is the projection of an affine slice of $\mathbb{R}_+^k$, where $k = 2n$.

- Smallest such k is the linear extension complexity of B_1^n

Extension complexity

Given a polytope

$$\begin{aligned} P &= \{(x_1, \ldots, x_n) \in \mathbb{R}^n : h_i(x) \geq 0 \ \text{ for } i \in [f]\} \\ &= \operatorname{conv}(p_1, \ldots, p_v). \end{aligned}$$

Extension complexity

Given a polytope

$$\begin{aligned} P &= \{(x_1, \ldots, x_n) \in \mathbb{R}^n : h_i(x) \geq 0 \ \text{ for } i \in [f]\} \\ &= \mathrm{conv}(p_1, \ldots, p_v). \end{aligned}$$

The slack matrix S_P of P is

Extension complexity

Given a polytope

$$
\begin{aligned}
P &= \{(x_1, \ldots, x_n) \in \mathbb{R}^n : h_i(x) \geq 0 \ \text{ for } i \in [f]\} \\
&= \operatorname{conv}(p_1, \ldots, p_v).
\end{aligned}
$$

The slack matrix S_P of P is

$$S_P = \big(h_j(p_i)\big) \in \mathbb{R}_+^{v \times f}$$

Extension complexity

Given a polytope

$$\begin{aligned} P &= \{(x_1, \ldots, x_n) \in \mathbb{R}^n : h_i(x) \geq 0 \ \text{ for } i \in [f]\} \\ &= \mathrm{conv}(p_1, \ldots, p_v). \end{aligned}$$

The slack matrix S_P of P is

$$S_P = \big(h_j(p_i)\big) \in \mathbb{R}_+^{v \times f}$$

Theorem (Yannakakis '91)

$\mathrm{rank}_+(S_P) \leq k \Leftrightarrow P$ *is the projection of an affine slice of* $\mathbb{R}_+^k$.

Extension complexity

Given a polytope

$$\begin{aligned} P &= \{(x_1, \ldots, x_n) \in \mathbb{R}^n : h_i(x) \geq 0 \ \text{ for } i \in [f]\} \\ &= \operatorname{conv}(p_1, \ldots, p_v). \end{aligned}$$

The slack matrix S_P of P is

$$S_P = \big(h_j(p_i)\big) \in \mathbb{R}_+^{v \times f}$$

Theorem (Yannakakis '91)

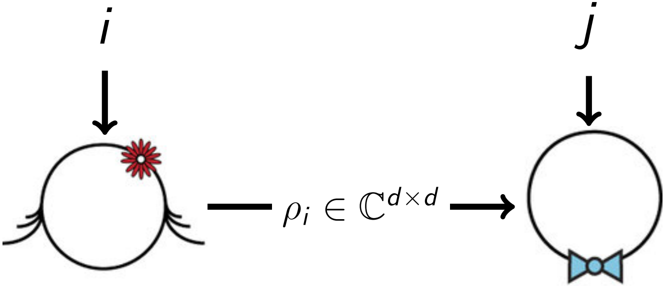
$\operatorname{rank}_+(S_P) \leq k \Leftrightarrow P$ *is the projection of an affine slice of* $\mathbb{R}_+^k$.

Theorem (Gouveia-Parrilo-Thomas '13)

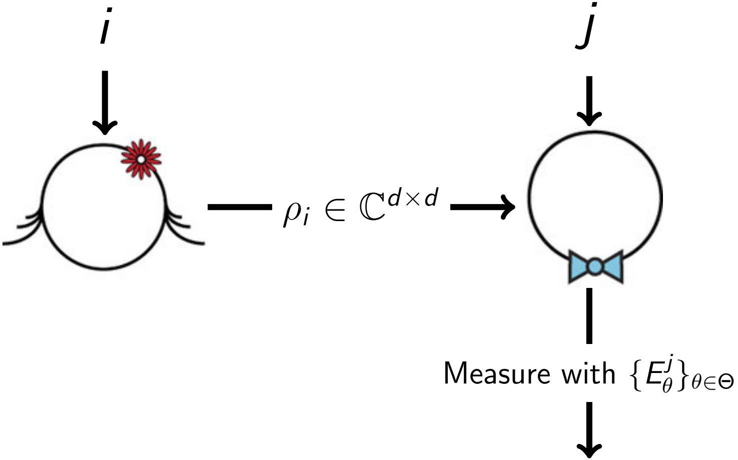
$\operatorname{rank}_{\mathrm{psd}}(S_P) \leq k \Leftrightarrow P$ *is the projection of an affine slice of* $\mathcal{S}_+^k$.

Quantum Communication complexity: A_{ij} in expectation

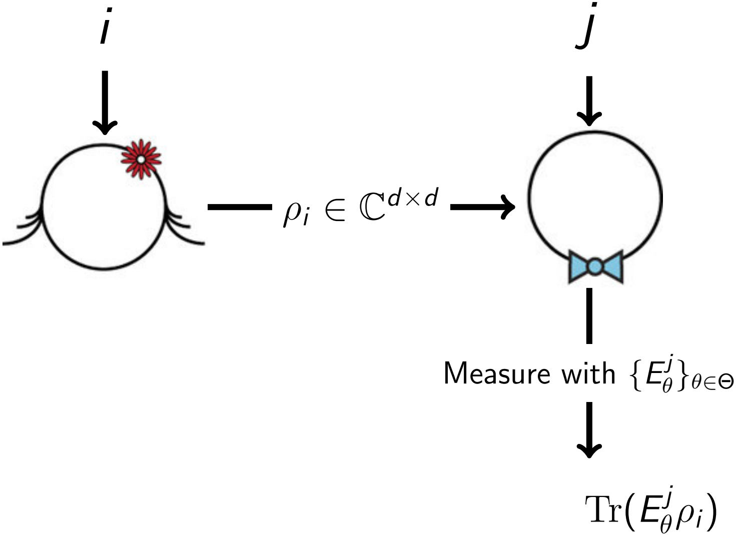
Quantum Communication complexity: A_{ij} in expectation



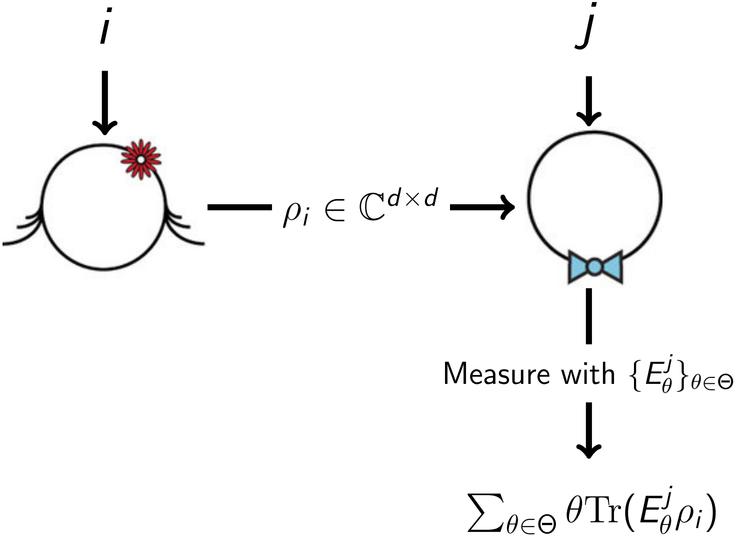
Quantum Communication complexity: A_{ij} in expectation



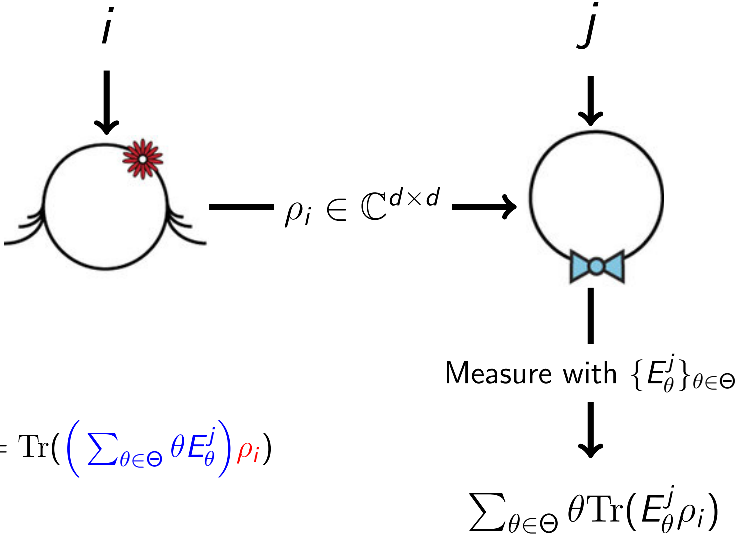
Quantum Communication complexity: A_{ij} in expectation



Quantum Communication complexity: A_{ij} in expectation

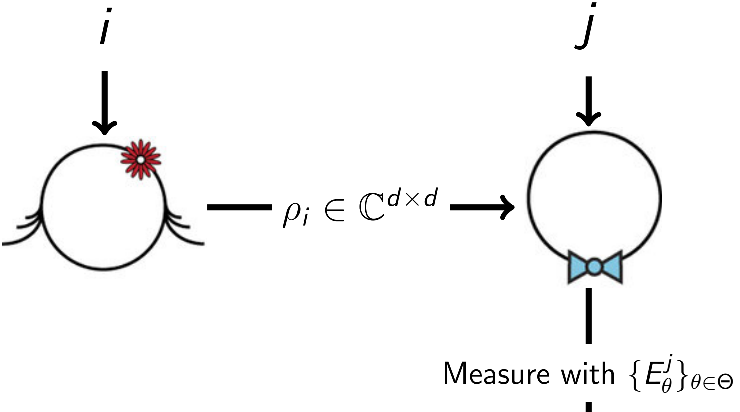


Quantum Communication complexity: A_{ij} in expectation



$$A_{ij} = \mathrm{Tr}\Big(\Big(\sum_{\theta \in \Theta} \theta E_\theta^j\Big) \rho_i\Big)$$

Quantum Communication complexity: A_{ij} in expectation



$$A_{ij} = \mathrm{Tr}\Big(\Big(\sum_{\theta\in\Theta} \theta E^j_\theta\Big)\rho_i\Big)$$

psd-rank$(A) \approx \min d$

$$\sum_{\theta\in\Theta} \theta \,\mathrm{Tr}(E^j_\theta \rho_i)$$

Clustering

$$\begin{pmatrix} \quad & \quad & \quad \\ & & \\ & & \\ & & \end{pmatrix} = \begin{pmatrix} \quad \\ \\ \\ \end{pmatrix} \begin{pmatrix} \quad & \quad & \quad \end{pmatrix}$$

Clustering

$$\text{data}\left\{\vphantom{\begin{pmatrix} \\ \\ \\ \\ \\ \end{pmatrix}}\right. \overbrace{\begin{pmatrix} \quad & \quad & \quad \\ \\ \\ \\ \\ \end{pmatrix}}^{\text{features}} = \begin{pmatrix} \quad \\ \\ \\ \\ \end{pmatrix} \begin{pmatrix} \quad & \quad & \quad \end{pmatrix}$$

Clustering

$$\text{data}\left\{\vphantom{\begin{pmatrix} \\ \\ \\ \\ \\ \end{pmatrix}}\right. \overbrace{\begin{pmatrix} \quad & \quad & \quad \\ \\ \\ \\ \\ \end{pmatrix}}^{\text{features}} = \overbrace{\begin{pmatrix} \quad \\ \\ \\ \\ \\ \end{pmatrix}}^{\text{k}} \begin{pmatrix} \quad & \quad & \quad & \quad \\ \\ \end{pmatrix}$$

Clustering

$$\text{data}\left\{\vphantom{\begin{pmatrix} \\ \\ \\ \\ \end{pmatrix}}\right. \overbrace{\begin{pmatrix} \quad & \quad & \quad \\ \\ \\ \\ \end{pmatrix}}^{\text{features}} = \overbrace{\begin{pmatrix} a_i^T \\ \\ \\ \\ \end{pmatrix}}^{k} \begin{pmatrix} \quad & \quad & \quad & \quad \end{pmatrix}$$

Clustering

$$\text{data}\left\{\vphantom{\begin{pmatrix}\\ \\ \\ \\ \\ \end{pmatrix}}\right.\overbrace{\begin{pmatrix} & & & \\ & & & \\ & & & \\ & & & \\ & & & \end{pmatrix}}^{\text{features}} = \overbrace{\begin{pmatrix} & \\ & \\ & \\ & \\ & \end{pmatrix}}^{k}\begin{pmatrix} & & & & \end{pmatrix}$$

$\Rightarrow$ assign i to cluster c corresponding to largest component of a_i

How to compute these ranks?

How to compute these ranks?

- Difficult (NP-hard)

How to compute these ranks?

- Difficult (NP-hard)
- Lower bounds via semidefinite programming

How to compute these ranks?

- Difficult (NP-hard)
- Lower bounds via semidefinite programming
  - (Noncommutative) polynomial optimization (with a twist)

How to compute these ranks?

- Difficult (NP-hard)
- Lower bounds via semidefinite programming
  - (Noncommutative) polynomial optimization (with a twist)

arXiv:1708.01573