線形マトロイド交叉問題に対する

高速な近似アルゴリズム

寺尾 樹哉 (京都大学数理解析研究所D2)

2025年度冬のLA @ 京都大学

2026年1月28日

線形マトロイド交叉

入力：2つの $r \times n$ 行列 M_1, M_2

出力：M_1とM_2の両方で線形独立となる最大サイズの列集合

e.g.

$$\begin{array}{cc} & \begin{matrix} ① & ② & ③ & ④ & ⑤ \end{matrix} \\ M_1 = & \begin{bmatrix} 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 \end{bmatrix} \end{array}$$

$$M_2 = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 1 & 0 \end{bmatrix}$$

線形マトロイド交叉

入力: 2つの $r \times n$ 行列 M_1, M_2

出力: M_1とM_2の両方で線形独立となる最大サイズの列集合

e.g.

$$M_1 = \begin{array}{ccccc} \text{①} & \text{②} & \text{③} & \text{④} & \text{⑤} \\ \hline 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 \end{array}$$

$$M_2 = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 1 & 0 \end{bmatrix}$$

線形マトロイド交叉

入力：2つの $r \times n$ 行列 M_1, M_2

出力：M_1とM_2の両方で線形独立となる最大サイズの列集合

e.g.

$$M_1 = \begin{array}{ccccc} ① & ② & ③ & ④ & ⑤ \\ \hline 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 \end{array}$$

$$M_2 = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 1 & 0 \end{bmatrix}$$

線形マトロイド交叉

入力：2つの $r \times n$ 行列 M_1, M_2

出力：M_1とM_2の両方で線形独立となる最大サイズの列集合

e.g.

$$\begin{array}{cc} & \begin{array}{ccccc} \textcircled{1} & \textcircled{2} & \textcircled{3} & \textcircled{4} & \textcircled{5} \end{array} \\ M_1 = & \left[\begin{array}{|ccc|cc} \hline 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 \\ \hline \end{array}\right] \end{array}$$

OK!

$$M_2 = \left[\begin{array}{|ccc|cc} \hline 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 1 & 0 \\ \hline \end{array}\right]$$

NG!

線形マトロイド交叉

入力：2つの $r \times n$ 行列 M_1, M_2

出力：M_1とM_2の両方で線形独立となる最大サイズの列集合

e.g.

$$M_1 = \begin{array}{ccccc} ① & ② & ③ & ④ & ⑤ \\ \hline 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 \end{array}$$

$$M_2 = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 1 & 0 \end{bmatrix}$$

線形マトロイド交叉

入力: 2つの $r \times n$ 行列 M_1, M_2

出力: M_1とM_2の両方で線形独立となる最大サイズの列集合

e.g.

$$M_1 = \begin{array}{ccccc} ① & ② & ③ & ④ & ⑤ \\ \hline 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 \end{array}$$

OK!

$$M_2 = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 1 & 0 \end{bmatrix}$$

OK!

線形マトロイド交叉はなぜ重要？

入力：2つの $r \times n$ 行列 M_1, M_2

出力：M_1 と M_2 の両方で線形独立となる最大サイズの列集合

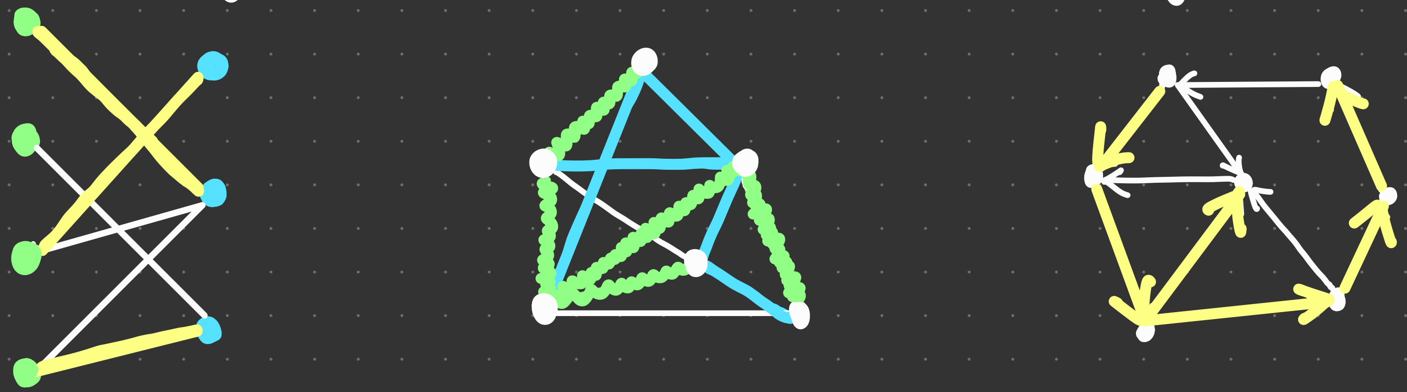
線形マトロイド交叉はなぜ重要？

入力：2つの $r \times n$ 行列 M_1, M_2

出力：M_1 と M_2 の両方で線形独立となる最大サイズの列集合

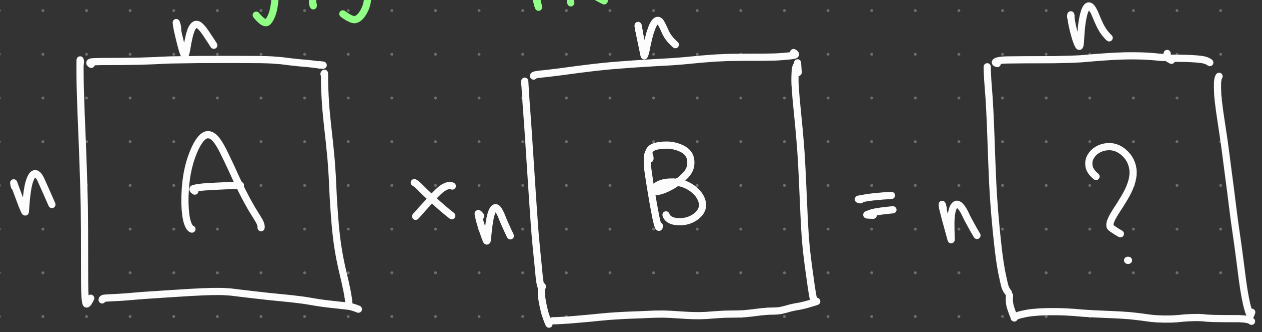
多くの重要な組合せ最適化問題を表現できる！

e.g. 二部グラフのマッチング, 全域木の詰め込み, 有向全域木



行列の問題に対する高速なアルゴリズム

- n×n行列同士の積

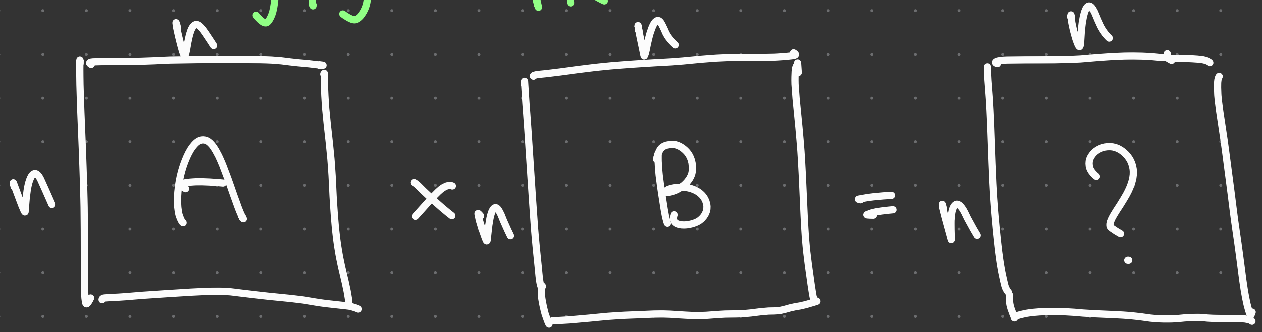


自明

$O(n^3)$ time

行列の問題に対する高速なアルゴリズム

- $n\times n$ 行列同士の積



自明	$O(n^3)$
Strassen '69	$O(n^{\log_2 7}) = O(n^{2.807})$
⋮	
Alman-Duan-V. Williams-Xu-Xu-Zhou '25	$O(n^{2.371339})$

行列の問題に対する高速なアルゴリズム

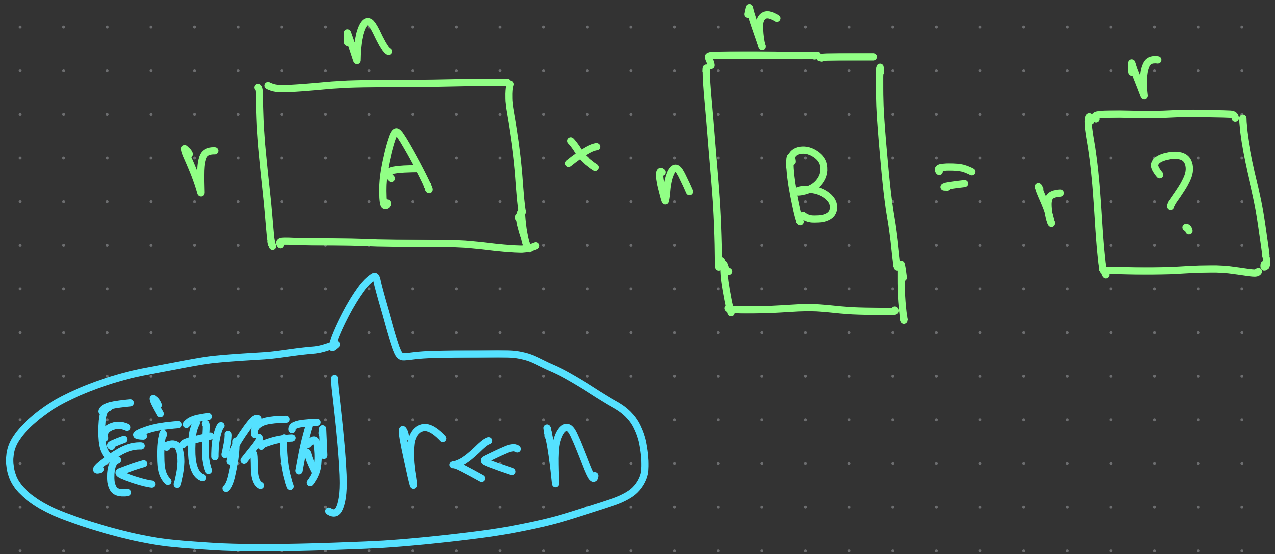
- $n \times n$ 行列同士の積：$O(n^{\omega})$ time　$(\omega < 2.372)$

行列の問題に対する高速なアルゴリズム

- $n \times n$ 行列同士の積 : $O(n^{\omega})$ time $(\omega < 2.372)$
- $n \times n$ 行列のランク, 逆行列の計算 : $O(n^{\omega})$ time

行列の問題に対する高速なアルゴリズム

- $n \times n$ 行列同士の積 : $O(n^{\omega})$ time $(\omega < 2.372)$
- $n \times n$ 行列のランク, 逆行列の計算 : $O(n^{\omega})$ time
- $r \times n$ 行列 と $n \times r$ 行列の積 : $O(nr^{\omega-1})$ time



$r \times n$行列Mのランクを計算する高速なアルゴリズム

$n \times n$行列同士の積: $O(n^{\omega})$

($\omega < 2.372$)

($r \times n$行列)$\times$($n \times r$行列): $O(nr^{\omega-1})$

$\bar{r} = \mathrm{rank}(M)$

自明(掃き出し法) : $O(nr\bar{r})$

n×n 行列同士の積: $O(n^{\omega})$
($\omega < 2.372$)
($r \times n$ 行列)×($n \times r$ 行列): $O(nr^{\omega-1})$

$r \times n$ 行列 M のランクを計算する高速なアルゴリズム

$\bar{r}$ = rank(M)

自明(掃き出し法) : $O(nr\bar{r})$

Buch-Hopcroft '74 : $O(nr^{\omega-1})$

$n \times n$ 行列同士の積: $O(n^{\omega})$
$(\omega < 2.372)$
($r \times n$ 行列)×($n \times r$ 行列): $O(nr^{\omega-1})$

$r \times n$ 行列 M のランク を計算する高速なアルゴリズム

$M = r\,\overset{n}{\boxed{\qquad\qquad}}$

$\bar{r} = \mathrm{rank}(M)$

自明(掃き出し法) : $O(nr\bar{r})$

Buch-Hopcroft '74 : $O(nr^{\omega-1})$

Storjohann '09 : $O(nr\bar{r}^{\omega-2})$

ランク $\bar{r}$ が小さければ速く解けるハズ

$r \times n$ 行列 M のランク を計算する高速なアルゴリズム

$n \times n$ 行列同士の積: $O(n^{\omega})$

$(\omega < 2.372)$

$(r \times n$ 行列$) \times (n \times r$ 行列$)$: $O(nr^{\omega-1})$

$$M = r\ \overset{n}{\begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 & 3 \\ 0 & 0 & 2 & 0 & 1 & 0 \end{bmatrix}}$$

自明(掃き出し法) : $O(nr\bar{r})$

Buch-Hopcroft '74 : $O(nr^{\omega-1})$

Storjohann '09 : $O(nr\bar{r}^{\omega-2})$

$\bar{r} = \mathrm{rank}(M)$, $\mathrm{nnz}(M) :=$ M の非ゼロ要素数

Cheung-Kwok-Lau '13 : $O(\mathrm{nnz}(M) + \bar{r}^{\omega})$

n×n 行列同士の積: $O(n^\omega)$
$(\omega < 2.372)$
(r×n 行列)×(n×r 行列): $O(nr^{\omega-1})$

r×n 行列 M のランク を計算する高速なアルゴリズム

$$M = r \overset{n}{\begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 & 3 \\ 0 & 0 & 2 & 0 & 1 & 0 \end{bmatrix}}$$

自明(掃き出し法) : $O(n r \bar{r})$

Bunch-Hopcroft '74 : $O(n r^{\omega-1})$

Storjohann '09 : $O(n r \bar{r}^{\omega-2})$

$\bar{r} = \mathrm{rank}(M)$, $\mathrm{nnz}(M)$:= M の非ゼロ要素数

Cheung-Kwok-Lau '13 : $O(\mathrm{nnz}(M) + \bar{r}^{\omega})$

行列Mが疎なら速く解けアリズ

$r \times n$ 行列 M のランクを計算する高速なアルゴリズム

$n \times n$ 行列同士の積: $O(n^{\omega})$

$(\omega < 2.372)$

$(r \times n$ 行列$) \times (n \times r$ 行列$)$: $O(nr^{\omega-1})$

$$M = \begin{array}{c|c} & n \\ \hline r & \begin{bmatrix} 1 & 2 & 1 & 1 & 1 & 5 & 2 \\ 2 & 1 & 1 & 1 & 2 & 6 & 1 \\ 3 & 1 & 2 & 3 & 4 & 3 & 4 \end{bmatrix} \end{array}$$

自明（掃き出し法） : $O(nr\bar{r})$

Bunch-Hopcroft '74 : $O(nr^{\omega-1})$

Storjohann '09 : $O(nr\bar{r}^{\omega-2})$

$\bar{r} = \mathrm{rank}(M)$, $\mathrm{nnz}(M) :=$ M の非ゼロ要素数

Cheung-Kwok-Lau '13 : $O(\underbrace{\mathrm{nnz}(M)}_{\le nr} + \bar{r}^{\omega})$

$r \times n$ 行列 M のランク

を計算する高速なアルゴリズム

$n \times n$ 行列同士の積: $O(n^{\omega})$

($\omega < 2.372$)

($r \times n$ 行列)×($n \times r$ 行列): $O(nr^{\omega-1})$

$\bar{r} = \mathrm{rank}(M)$, $\mathrm{nnz}(M) :=$ M の非ゼロ要素数

Cheung-Kwok-Lau '13 : $O(\underbrace{\mathrm{nnz}(M)}_{\text{入力サイズ}} + \bar{r}^{\omega})$

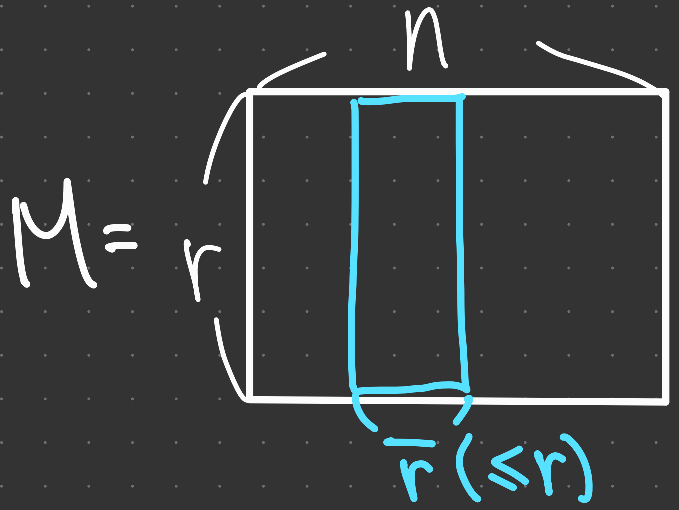
r×n 行列 M のランク を計算する高速なアルゴリズム

n×n 行列同士の積: $O(n^\omega)$
$(\omega < 2.372)$
(r×n 行列)×(n×r 行列): $O(nr^{\omega-1})$

$\bar{r} = \mathrm{rank}(M)$, $\mathrm{nnz}(M)$:= M の非ゼロ要素数

Cheung-Kwok-Lau '13 : $O(\underbrace{\mathrm{nnz}(M)}_{} + \bar{r}^\omega)$

入力サイズ



n×n 行列同士の積: $O(n^\omega)$
($\omega < 2.372$)
($r\times n$ 行列)×($n\times r$ 行列): $O(nr^{\omega-1})$

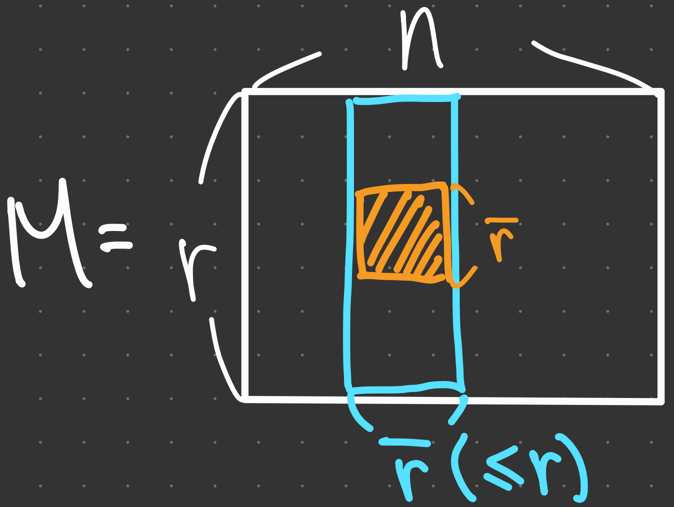
$r\times n$ 行列 M のランクを計算する高速なアルゴリズム

$\bar{r}$ = rank(M), nnz(M) := Mの非ゼロ要素数

Cheung-Kwok-Lau '13 : $O(\mathrm{nnz}(M) + \bar{r}^{\omega})$

入力サイズ

$\bar{r}\times\bar{r}$ 行列が full rank か確かめるのに必要



$n\times n$行列同士の積: $O(n^{\omega})$
$(\omega < 2.372)$
$(r\times n$行列$)\times(n\times r$行列$)$: $O(nr^{\omega-1})$

$r\times n$行列Mのランクを計算する高速なアルゴリズム

$\bar{r} = \mathrm{rank}(M)$, $\mathrm{nnz}(M) :=$ Mの非ゼロ要素数

Cheung-Kwok-Lau '13 : $O(\mathrm{nnz}(M) + \underline{\underline{\bar{r}^{\omega}}})$

$\mathrm{nnz}(M)$ ← 入力サイズ

$\bar{r}^{\omega}$ ← $\bar{r}\times\bar{r}$行列がfull rankか確かめるのに必要

⇒ "行列の疎さ"と"$\bar{r}$の小ささ"を限界まで使った計算量!

$n \times n$ 行列同士の積: $O(n^{\omega})$
$(\omega < 2.372)$
$(r \times n$ 行列$) \times (n \times r$ 行列$)$: $O(nr^{\omega-1})$

最大重み独立集合 を計算する高速なアルゴリズム

入力: 行列 $M = r\,\overset{\textcircled{1}\textcircled{2}\cdots\textcircled{n}}{\boxed{\qquad\qquad}}$, 重み $c: \{\textcircled{1}, \textcircled{2}, \ldots, \textcircled{n}\} \to \mathbb{R}_{\geq 0}$

出力: 線形独立な
$\sum_{\textcircled{i} \in I} c(\textcircled{i})$ が最大の列集合 $I \subseteq \{\textcircled{1}, \textcircled{2}, \ldots, \textcircled{n}\}$

(r×n行列)×(n×r行列): $O(nr^{\omega-1})$
行列Mのランクの計算:
[Cheung-Kwok-Lau'13] $O(\mathrm{nnz}(M)+\bar{r}^{\omega})$

最大重み独立集合
を計算する高速なアルゴリズム

入力: 行列 $M = \begin{array}{c} ①②\cdots ⓝ \\ r\,\boxed{\qquad\qquad} \end{array}$, 重み $c: \{①,②,\cdots,ⓝ\} \to \mathbb{R}_{\geq 0}$

出力: 線形独立な
$\sum_{ⓘ\in I} c(ⓘ)$ が最大の列集合 $I \subseteq \{①,②,\cdots,ⓝ\}$

Cheung-Kwok-Lau'13

$\bar{r} = \mathrm{rank}(M)$

$\tilde{O}(\mathrm{nnz}(M) + n\bar{r}^{\omega-1})$

最大重み独立集合

を計算する高速なアルゴリズム

$(r\times n$ 行列$)\times(n\times r$ 行列$)$: $O(nr^{\omega-1})$

行列 M のランクの計算:

[Cheung-Kwok-Lau'13] $O(\mathrm{nnz}(M)+\bar{r}^{\omega})$

入力: 行列 $M = \begin{array}{cc} & ①②\cdots ⓝ \\ r & \boxed{\quad\quad\quad} \end{array}$, 重み $c:\{①,②,\cdots,ⓝ\}\to\mathbb{R}_{\geq 0}$

出力: 線形独立な

$\sum_{ⓘ\in I} c(ⓘ)$ が最大の列集合 $I\subseteq\{①,②,\cdots,ⓝ\}$

$\bar{r} = \mathrm{rank}(M)$

Cheung-Kwok-Lau'13 $\tilde{O}(\mathrm{nnz}(M)+n\bar{r}^{\omega-1})$

M を $\bar{r}\,\overset{n}{\boxed{\quad\quad\quad}}$ に圧縮

$(r\times n \text{行列})\times(n\times r \text{行列})$: $O(nr^{\omega-1})$

行列 M のランクの計算:

[Cheung-Kwok-Lau'13] $O(\mathrm{nnz}(M)+\bar{r}^{\omega})$

最大重み独立集合 を計算する高速なアルゴリズム

入力: 行列 $M = r\begin{array}{c} ①②\cdots ⓝ \\ \boxed{} \end{array}$, 重み $c: \{①,②,\cdots,ⓝ\} \to \mathbb{R}_{\geq 0}$

出力: 線形独立な $\sum_{ⓘ\in I} c(ⓘ)$ が最大の列集合 $I \subseteq \{①,②,\cdots,ⓝ\}$

$\bar{r} = \mathrm{rank}(M)$

Cheung-Kwok-Lau'13 $\tilde{O}(\mathrm{nnz}(M) + n\bar{r}^{\omega-1})$

Dumas-Pernet-Sultan'17, Nguyễn'19 $(\mathrm{nnz}(M) + \bar{r}^{\omega})^{1+o(1)}$

(r×n行列)×(n×r行列): $O(nr^{\omega-1})$

行列Mのランクの計算:

[Cheung-Kwok-Lau'13] $O(\mathrm{nnz}(M)+\bar{r}^{\omega})$

最大重み独立集合

を計算する高速なアルゴリズム

入力: 行列 $M = r\overset{①②\cdots ⓝ}{\boxed{\qquad\qquad}}$, 重み $c: \{①,②,\cdots,ⓝ\} \to \mathbb{R}_{\geq 0}$

出力: 線形独立な $\sum_{ⓘ \in I} c(ⓘ)$ が最大の列集合 $I \subseteq \{①,②,\cdots,ⓝ\}$

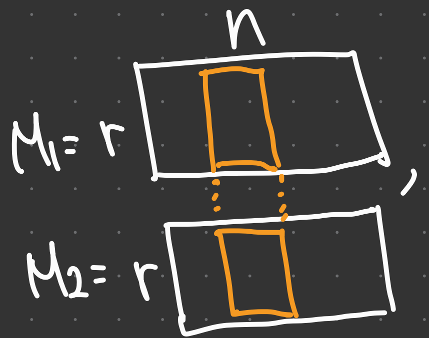
$\bar{r} = \mathrm{rank}(M)$

Cheung-Kwok-Lau'13 $\tilde{O}(\mathrm{nnz}(M)+n\bar{r}^{\omega-1})$

Dumas-Penet-Sultan'17, Nguyễn'19 $(\mathrm{nnz}(M)+\bar{r}^{\omega})^{1+o(1)}$

“Mの疎さ”と“$\bar{r}$の小ささ”を限界まで使った計算量

入力:



出力：両方で線形独立な最大サイズの列集合

$r_* :=$ 最適解のサイズ

Q. 線形マトロイド交叉問題を

$\tilde{O}(\mathrm{nnz}(M_1) + \mathrm{nnz}(M_2) + r_*^{\omega})$ time

で解ける？

行列の疎さと r_* の小ささを限界まで使った計算量！

線形マトロイド交叉の計算量

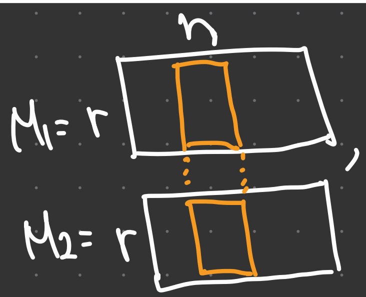
$n\times n$行列同士の積: $O(n^\omega)$ $(\omega<2.372)$

$(r\times n$行列$)\times(n\times r$行列$)$: $O(nr^{\omega-1})$

Edmonds '70

多項式時間

入力:



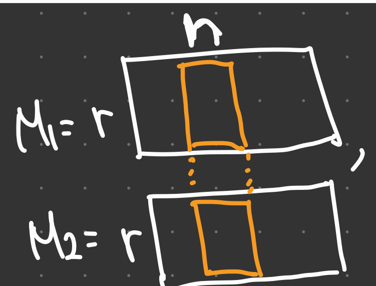
出力: 両方で線形独立な最大サイズの列集合

線形マトロイド交叉の計算量

$n \times n$ 行列同士の積：$O(n^{\omega})$ $(\omega < 2,372)$

($r \times n$ 行列) × ($n \times r$ 行列)；$O(nr^{\omega-1})$

入力：



出力：両方で線形独立な最大サイズの列集合

$r_* :=$ 最適解のサイズ

Edmonds '70	多項式時間
Cunningham '86	$\tilde{O}(nr^2)$
Gabow - Xu '89	$\tilde{O}(nr\,r_*^{1/(4-\omega)}) = \tilde{O}(nr\,r_*^{0.62})$
Harvey '06	$\tilde{O}(nr^{\omega-1})$

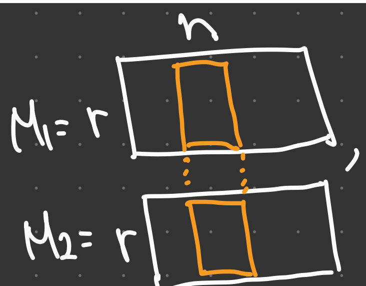
代数的乱択アルゴリズム

線形マトロイド交叉の計算量

$n\times n$ 行列同士の積: $O(n^{\omega})$ $(\omega<2.372)$

$(r\times n$ 行列$)\times(n\times r$ 行列$)$: $O(nr^{\omega-1})$

入力:



出力: 両方で線形独立な最大サイズの列集合

$r_* :=$ 最適解のサイズ

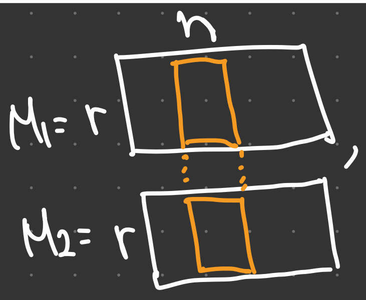
Edmonds '70	多項式時間
Cunningham '86	$\tilde{O}(nr^2)$
Gabow-Xu '89	$\tilde{O}(nr\, r_*^{1/(4-\omega)}) = \tilde{O}(nr\, r_*^{0.62})$
Harvey '06	$\tilde{O}(nr^{\omega-1})$
Cheung-Kwok-Lau '13	$\tilde{O}(\mathrm{nnz}(M_1) + \mathrm{nnz}(M_2) + nr_*^{\omega-1})$

M_1, M_2 を r_*  に圧縮!

線形マトロイド交叉の計算量

$n \times n$ 行列同士の積：$O(n^{\omega})$ $(\omega<2.372)$
($r \times n$ 行列) × ($n \times r$ 行列)：$O(nr^{\omega-1})$

入力：



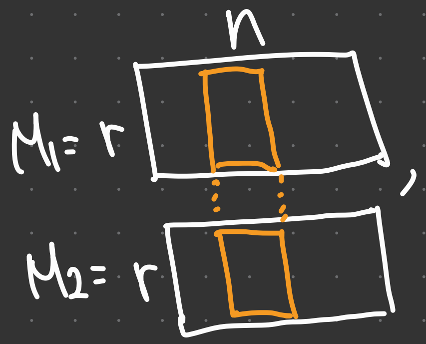
出力：両方で線形独立な最大サイズの列集合

$r_* :=$ 最適解のサイズ

Edmonds '70	厳密解	多項式時間
Cunningham '86		$\tilde{O}(nr^2)$
Gabow-Xu '89		$\tilde{O}(nr\,r_*^{1/(4-\omega)}) = \tilde{O}(nr\,r_*^{0.62})$
Harvey '06		$\tilde{O}(nr^{\omega-1})$
Cheung-Kwok-Lau '13		$\tilde{O}(\mathrm{nnz}(M_1)+\mathrm{nnz}(M_2)+nr_*^{\omega-1})$
本研究	$(1-\varepsilon)$ 近似	$\tilde{O}_{\varepsilon}(\mathrm{nnz}(M_1)+\mathrm{nnz}(M_2)+r_*^{\omega})$

サイズ $(1-\varepsilon)\cdot r_*$ 以上の解を出力

入力:



出力: 両方で線形独立な最大サイズの列集合

$r_* :=$ 最適解のサイズ

Q. 線形マトロイド交叉問題を

$$\tilde{O}(\mathrm{nnz}(M_1) + \mathrm{nnz}(M_2) + r_*^{\omega})$$

time で解ける?

行列の疎さと r_* の小ささを限界まで使った計算量!

A. $(1-\varepsilon)$-近似なら可能!

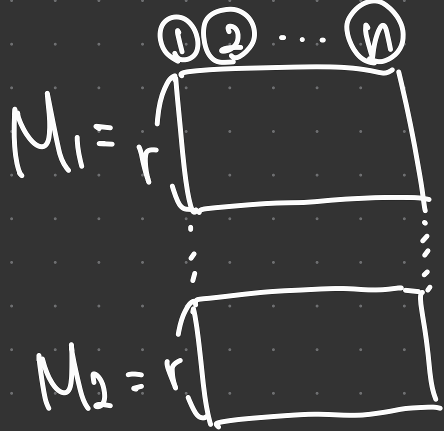
サイズ $(1-\varepsilon) \cdot r_*$ 以上の解を出力

n×n行列同士の積: $O(n^{\omega})$ $(\omega < 2.372)$

(r×n行列)×(n×r行列): $O(nr^{\omega-1})$

重み付き線形マトロイド交叉の計算量

入力:



, 重み $c: \{①, \dots, ⓝ\} \to \mathbb{R}_{\geq 0}$

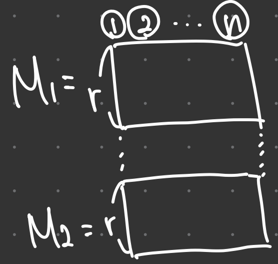
出力: 両方で線形独立な $\sum_{ⓘ \in I} c(ⓘ)$ が最大の列集合 $I \subseteq \{①, \dots, ⓝ\}$

重み付き線形マトロイド交叉の計算量

$n\times n$ 行列同士の積：$O(n^{\omega})$ $(\omega<2.372)$

($r\times n$ 行列)×($n\times r$ 行列)：$O(nr^{\omega-1})$

入力：



, 重み $c:\{①,\dots,ⓝ\}\to\mathbb{R}_{\geq 0}$

出力：両方で線形独立な $\sum_{ⓘ\in I} c(ⓘ)$ が最大の列集合 $I\subseteq\{①,\dots,ⓝ\}$

$r_*=$ 重みなしの最適解のサイズ

Huang-Kakimura-Kamiyama'16

(1-ε)近似 $\tilde{O}_{\varepsilon}(\mathrm{nnz}(M_1)+\mathrm{nnz}(M_2)+nr_*^{\omega-1})$

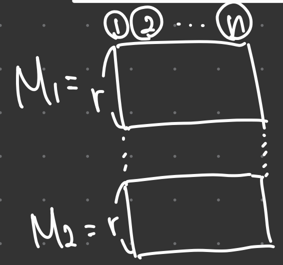
重み (1-ε)·OPT以上の解を出力

重み付き線形マトロイド交叉の計算量

$n \times n$ 行列同士の積: $O(n^{\omega})$ $(\omega < 2.372)$

$(r \times n$ 行列$) \times (n \times r$ 行列$)$: $O(nr^{\omega-1})$

入力:



, 重み $c: \{①, \dots, ⓝ\} \to \mathbb{R}_{\geq 0}$

出力: 両方で線形独立な $\sum_{ⓘ \in I} c(ⓘ)$ が最大の列集合 $I \subseteq \{①, \dots, ⓝ\}$

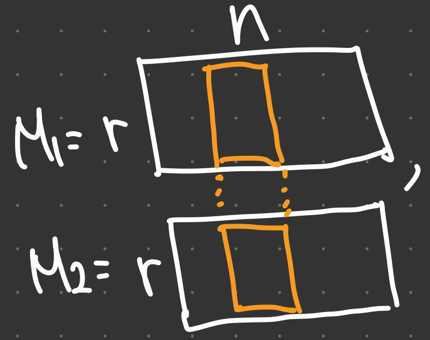
r_* = 重みなしの最適解のサイズ

Huang-Kakimura-Kamiyama'16 $(1-\varepsilon)$近似 $\tilde{O}_{\varepsilon}(\mathrm{nnz}(M_1) + \mathrm{nnz}(M_2) + nr_*^{\omega-1})$

本研究 $(1-\varepsilon)$近似 $\tilde{O}_{\varepsilon}(\mathrm{nnz}(M_1) + \mathrm{nnz}(M_2) + r_*^{\omega})$

線形マトロイド交叉

入力:



出力: 両方で線形独立な最大サイズの列集合

$r_* :=$ 最適解のサイズ

サイズ$(1-\varepsilon)r_*$以上の解を出力

本研究

- $(1-\varepsilon)$近似
- $\tilde{O}_\varepsilon(\mathrm{nnz}(M_1)+\mathrm{nnz}(M_2)+r_*^\omega)$ time

アイデア

Quanrud'24のオラクルモデルでのアルゴリズム + spanを高速に計算するアルゴリズム

Quanrud'24 のマトロイド交叉アルゴリズム

入力：2つのマトロイド $M_1=(V,\mathcal{I}_1), (V,\mathcal{I}_2)$

出力：最大サイズの $I\in\mathcal{I}_1\cap\mathcal{I}_2$

$n=|V|,\ r_*=|I|$

```
① $w(e) \leftarrow 1$ ( for $e\in V$ )      $L=\Theta(\log(n)/\varepsilon)$
② for $\ell=1$ to $L$ do
    (a) サイズ $\Theta(r_*)$ の集合 $V'\subseteq V$ をサンプル
          (各要素 $e\in V$ は $w(e)$の比率の確率で独立にとる)
    (b) $V'$ 上で primal sol. $I^{(\ell)}$
                  dual sol. $(\tilde{S}^{(\ell)}, \tilde{T}^{(\ell)})$ を計算
    (c) $S^{(\ell)}=\mathrm{span}_{M_1}(\tilde{S}^{(\ell)})$, $T^{(\ell)}=\mathrm{span}_{M_2}(\tilde{T}^{(\ell)})$
    (d) for $e\in S^{(\ell)}\cup T^{(\ell)}$, $w(e)\leftarrow w(e)/2$
③ $|I^{(\ell)}|$ が最大となる $I^{(\ell)}$ を出力
```

Quanrud'24 のマトロイド交叉アルゴリズム

入力: 2つのマトロイド
$M_1=(V,\mathcal{I}_1), (V,\mathcal{I}_2)$
出力: 最大サイズの $I\in\mathcal{I}_1\cap\mathcal{I}_2$
$n=|V|,\ r_*=|I|$

```
① $w(e) \leftarrow 1$ ( for $e\in V$ )
② for $\ell=1$ to $L$ do
    (a) サイズ $\tilde{\Theta}(r_*)$ の集合 $V'\subseteq V$ をサンプル
        (各要素 $e\in V$ は $w(e)$ の比率の確率で独立にとる)
    (b) $V'$ 上で primal sol. $I^{(\ell)}$
                 dual sol. $(\tilde{S}^{(\ell)}, \tilde{T}^{(\ell)})$ を計算
    (c) $S^{(\ell)} = \mathrm{span}_{M_1}(\tilde{S}^{(\ell)}),\ T^{(\ell)} = \mathrm{span}_{M_2}(\tilde{T}^{(\ell)})$
    (d) for $e\in S^{(\ell)}\cup T^{(\ell)}$,  $w(e)\leftarrow w(e)/2$
③ $|I^{(\ell)}|$ が最大となる $I^{(\ell)}$ を出力
```

$L=\Theta(\log(n)/\varepsilon)$

線形マトロイド交叉

目標
$\tilde{O}_\varepsilon(\mathrm{nnz}(M_1)+\mathrm{nnz}(M_2)+r_*^\omega)$

Harvey'06 で
$\tilde{O}(r_*^\omega)$ time

Quanrud'24 のマトロイド交叉アルゴリズム

入力: 2つのマトロイド
$\mathcal{M}_1=(V,\mathcal{I}_1), (V,\mathcal{I}_2)$
出力: 最大サイズの $I \in \mathcal{I}_1 \cap \mathcal{I}_2$
$n=|V|, r_*=|I|$

```
① $w(e) \leftarrow 1$ ( for $e \in V$ )
② for $\ell = 1$ to $L$ do
    (a) サイズ $\tilde{\Theta}(r_*)$ の集合 $V' \subseteq V$ をサンプル
        (各要素 $e \in V$ は $w(e)$ の比率の確率で独立にとる)
    (b) $V'$ 上で primal sol. $I^{(\ell)}$
                 dual sol. $(\tilde{S}^{(\ell)}, \tilde{T}^{(\ell)})$ を計算
    (c) $S^{(\ell)} = \mathrm{span}_{\mathcal{M}_1}(\tilde{S}^{(\ell)})$, $T^{(\ell)} = \mathrm{span}_{\mathcal{M}_2}(\tilde{T}^{(\ell)})$
    (d) for $e \in S^{(\ell)} \cup T^{(\ell)}$, $w(e) \leftarrow w(e)/2$
③ $|I^{(\ell)}|$ が最大となる $I^{(\ell)}$ を出力
```

$L = \Theta(\log(n)/\varepsilon)$

線形マトロイド交叉

目標
$\tilde{O}_\varepsilon(\mathrm{nnz}(M_1) + \mathrm{nnz}(M_2) + r_*^\omega)$

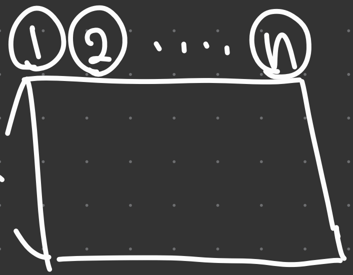
Harvey'06 で
$\tilde{O}(r_*^\omega)$ time

この部分の計算量は?

タスク：$\mathrm{span}_M(S)$ を高速に計算

$n \times n$ 行列同士の積：$O(n^{\omega})$ $(\omega < 2.372)$

$(r \times n \text{ 行列}) \times (n \times r \text{ 行列})$：$O(nr^{\omega-1})$

入力：$r \times n$ 行列 $M =$ 

，$S \subseteq \{$①，②，…，ⓝ$\}$

出力：$\mathrm{span}_M(S)$

$$= \{\, \text{ⓘ} \in \{\text{①},\dots,\text{ⓝ}\} \mid \mathrm{rank}(M[S]) = \mathrm{rank}(M[S \cup \{\text{ⓘ}\}]) \,\}$$

$n\times n$行列同士の積: $O(n^{\omega})$ $(\omega<2,372)$

$(r\times n$行列$)\times(n\times r$行列$)$: $O(nr^{\omega-1})$

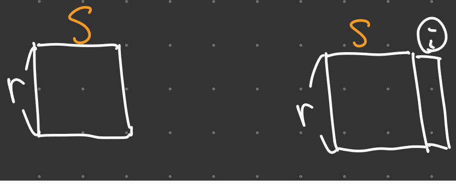
タスク: $\mathrm{span}_M(S)$ を高速に計算

入力: $r\times n$行列 $M =$ 

, $S \subseteq \{①,②,\dots,ⓝ\}$

出力: $\mathrm{span}_M(S)$

$= \{ⓘ \in \{①,\dots,ⓝ\} \mid \mathrm{rank}(M[S]) = \mathrm{rank}(M[S\cup\{ⓘ\}])\}$



自明 $O(nr^2)$

(掃き出し法)

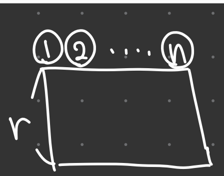
有名事実を使うと… $O(nr^{\omega-1})$

タスク: $\mathrm{span}_M(S)$ を高速に計算

$n\times n$行列同士の積: $O(n^{\omega})$ $(\omega<2.372)$

$(r\times n$行列$)\times(n\times r$行列$)$: $O(nr^{\omega-1})$

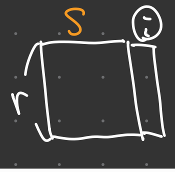
入力: $r\times n$行列 $M=$



, $S\subseteq\{①,②,\dots,ⓝ\}$

出力: $\mathrm{span}_M(S)$

$=\{ⓘ\in\{①,\dots,ⓝ\}\mid \mathrm{rank}(M[S])=\mathrm{rank}(M[S\cup\{ⓘ\}])\}$



自明 $O(nr^2)$

(掃き出し法)

有名事実を使うと… $O(nr^{\omega-1})$

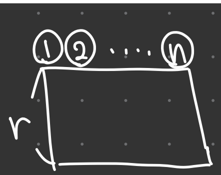
$(r\times n$行列$)\times(n\times r$行列$)$を使わないといけない!

タスク: $\mathrm{span}_M(S)$ を高速に計算

$n \times n$ 行列同士の積: $O(n^\omega)$ $(\omega < 2.372)$

$(r \times n$ 行列$) \times (n \times r$ 行列$)$: $O(nr^{\omega-1})$

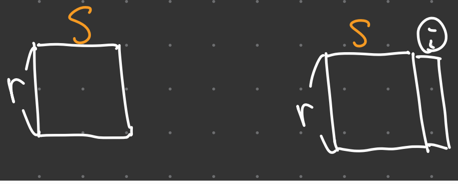
入力: $r \times n$ 行列 $M =$



, $S \subseteq \{①, ②, \dots, ⓝ\}$

出力: $\mathrm{span}_M(S)$

$= \{ ⓘ \in \{①, \dots, ⓝ\} \mid \mathrm{rank}(M[S]) = \mathrm{rank}(M[S \cup \{ⓘ\}]) \}$



自明 $O(nr^2)$

(掃き出し法)

有名事実を使うと… $O(nr^{\omega-1})$

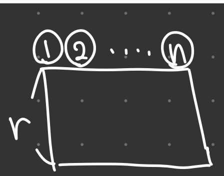
(r×n行列)×(n×r行列)を使わないといけない!

目標 $\tilde{O}(\mathrm{nnz}(M) + r^\omega)$

タスク: $\mathrm{span}_M(S)$ を高速に計算

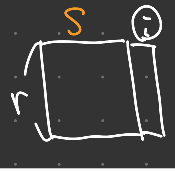
n×n行列同士の積: $O(n^\omega)$ $(\omega < 2.372)$

(r×n行列)×(n×r行列): $O(nr^{\omega-1})$

入力: r×n行列 $M =$  , $S \subseteq \{\textcircled{1}, \textcircled{2}, \dots, \textcircled{n}\}$

出力: $\mathrm{span}_M(S)$

$$= \{\textcircled{i} \in \{\textcircled{1}, \dots, \textcircled{n}\} \mid \mathrm{rank}(M[S]) = \mathrm{rank}(M[S \cup \{\textcircled{i}\}])\}$$



自明 $O(nr^2)$

(掃き出し法)

有名事実を使うと… $O(nr^{\omega-1})$

(r×n行列)×(n×r行列)を使わないといけない!

目標 $\tilde{O}(\mathrm{nnz}(M) + r^\omega)$

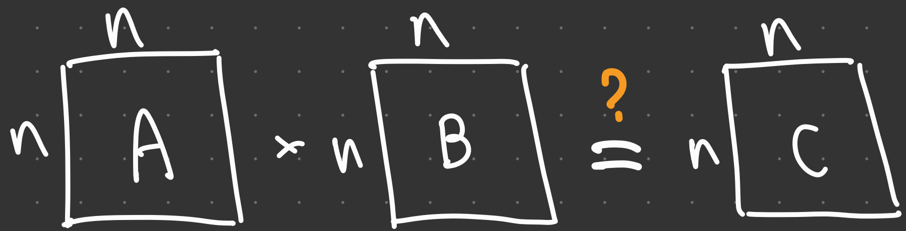
(r×n行列)×(n×r行列)を回避しないといけない! でも、どうすれば??

n×n行列同士の積：$O(n^{\omega})$ $(\omega<2.373)$

ヒント：Freivalds' アルゴリズム

入力：n×n行列 A, B, C

問題：$A \times B = C$ か判定

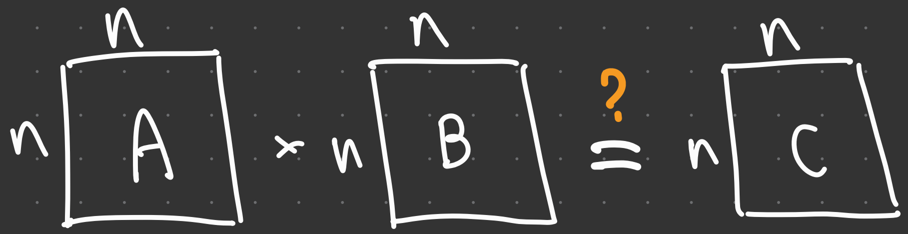


n×n行列同士の積: $O(n^{\omega})$ $(\omega < 2.372)$

ヒント: Freivalds' アルゴリズム

入力: n×n行列 A, B, C

問題: A×B = C か判定



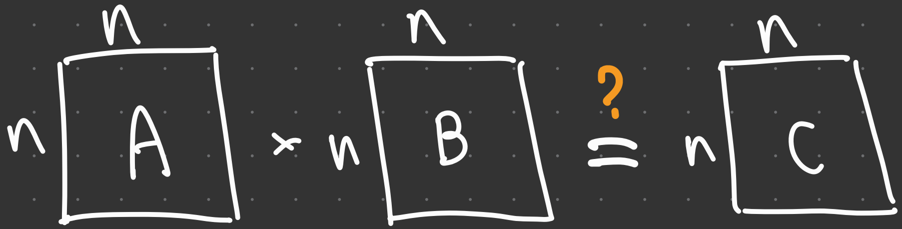
計算量　自明　$O(n^{\omega})$

n×n行列同士の積: $O(n^{\omega})$ $(\omega<2.373)$

ヒント: Freivalds' アルゴリズム

入力: n×n行列 A, B, C

問題: $A \times B = C$ か判定



計算量

自明	$O(n^{\omega})$
Freivalds'79	$\tilde{O}(n^2)$

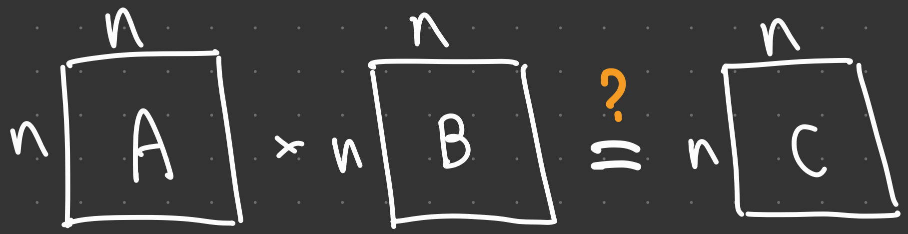
乱択アルゴリズム

n×n行列同士の積: $O(n^{\omega})$ $(\omega<2.373)$

ヒント: Freivalds' アルゴリズム

入力: n×n行列 A, B, C

問題: $A\times B=C$ か判定



計算量

自明	$O(n^{\omega})$
Freivalds'79	$\tilde{O}(n^2)$

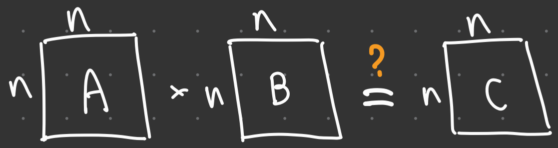
n×n行列同士の積を回避!

乱択アルゴリズム

n×n行列同士の積: $O(n^{\omega})$ $(\omega<2.373)$

入力: n×n行列 A, B, C

問題: $A\times B = C$ か判定



ヒント: Freivalds' アルゴリズム

乱択 $\tilde{O}(n^2)$ time

① ランダムな n次元ベクトル $\mathbb{r} \in \{0, 1, \dots, R-1\}^n$

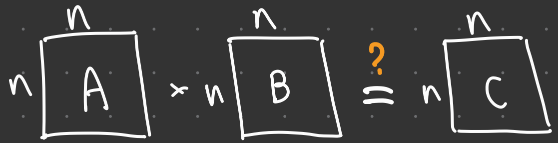
ヒント: Freivalds' アルゴリズム

乱択 $\tilde{O}(n^2)$ time

n×n行列同士の積: $O(n^{\omega})$ $(\omega<2.373)$

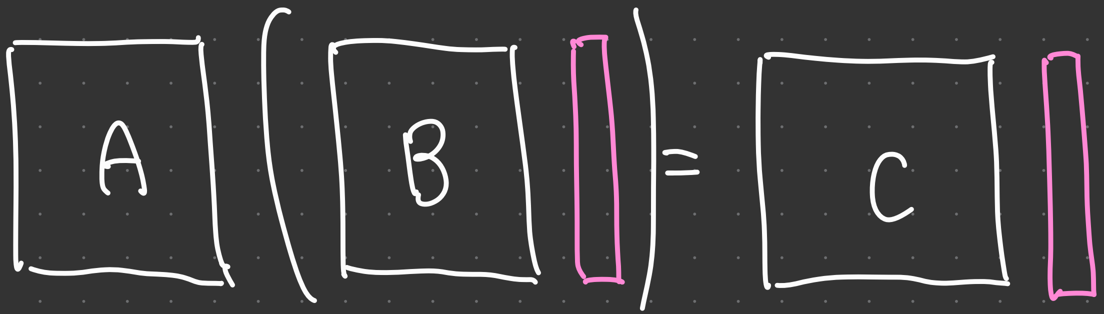
入力: n×n行列 A, B, C

問題: $A \times B = C$ か判定



① ランダムな n次元ベクトル $\mathbf{r} \in \{0, 1, \cdots, R-1\}^n$

② $A(B\mathbf{r}) = C\mathbf{r}$ ならば Yes, otherwise No.



$O(n^2)$ timeで計算できる!

n×n行列同士の積: $O(n^{\omega})$ $(\omega < 2.373)$

入力: n×n行列 A, B, C

問題: $A \times B = C$ か判定

$$\overset{n}{n\boxed{A}} \times \overset{n}{n\boxed{B}} \overset{?}{=} \overset{n}{n\boxed{C}}$$

ヒント: Freivalds' アルゴリズム

乱択 $\tilde{O}(n^2)$ time

① ランダムな n次元ベクトル $\mathbb{r} \in \{0, 1, \cdots, R-1\}^n$

② $A(B\mathbb{r}) = C\mathbb{r}$ ならば Yes, otherwise No.

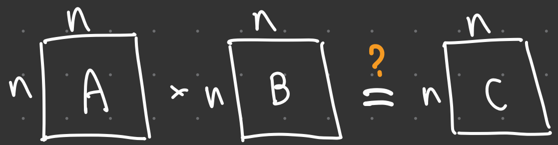
略証)

- $AB = C$ なら、必ず Yes を返す

n×n行列同士の積: $O(n^{\omega})$ $(\omega < 2.373)$

入力: n×n行列 A, B, C

問題: $A \times B = C$ か判定



ヒント: Freivalds' アルゴリズム

乱択 $\tilde{O}(n^2)$ time

① ランダムな n次元ベクトル $\mathbf{r} \in \{0, 1, \cdots, R-1\}^n$

② $A(B\mathbf{r}) = C\mathbf{r}$ ならば Yes, otherwise No.

略証)
- $AB = C$ なら、必ず Yes を返す
- $AB \neq C$ なら、確率 $1 - \frac{1}{R}$ 以上で No を返す

New：高速に $\mathrm{span}_M(S)$ を計算

$n \times n$ 行列同士の積：$O(n^{\omega})$ $(\omega < 2.372)$

$(r \times n$ 行列$) \times (n \times r$ 行列$)$：$O(nr^{\omega-1})$

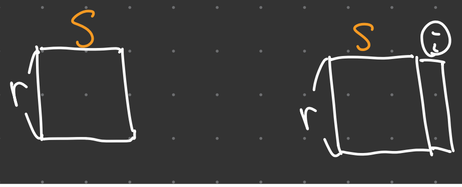
乱択 $\underline{\tilde{O}(\mathrm{nnz}(M) + r^{\omega})}$ $\underline{\text{time}}$

入力：$r \times n$ 行列 $M =$

, $S \leq \{①, ②, \dots, ⓝ\}$

出力：$\mathrm{span}_M(S)$

$= \{ ⓘ \in \{①, \dots, ⓝ\} \mid \mathrm{rank}(M[S]) = \mathrm{rank}(M[S \cup \{ⓘ\}]) \}$



New：高速に $\mathrm{span}_M(S)$ を計算

$n\times n$ 行列同士の積：$O(n^{\omega})$ $(\omega<2,372)$

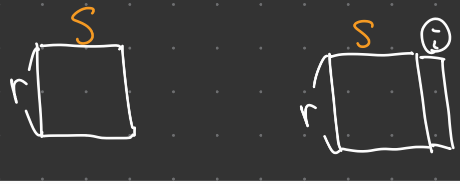
$(r\times n$ 行列$)\times(n\times r$ 行列$)$：$O(nr^{\omega-1})$

入力：$r\times n$ 行列 $M = r$

, $S \subseteq \{①,②,\dots,ⓝ\}$

出力：$\mathrm{span}_M(S)$

$= \{ⓘ \in \{①,\dots,ⓝ\} \mid \mathrm{rank}(M[S]) = \mathrm{rank}(M[S\cup\{ⓘ\}])\}$



乱択 $\tilde{O}(\mathrm{nnz}(M)+r^{\omega})$ time

$W := M[S]$ の列空間

① W の直交補空間 $W^{\perp} := \{v \in \mathbb{F}^{r} \mid {}^{\forall} w \in W, v\cdot w = 0\}$

から ランダムなベクトル r をサンプル

New : 高速に $\mathrm{span}_M(S)$ を計算

$n \times n$ 行列同士の積 : $O(n^{\omega})$ $(\omega < 2.372)$

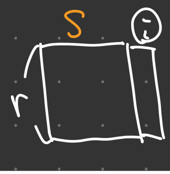
($r \times n$ 行列) × ($n \times r$ 行列) : $O(nr^{\omega-1})$

入力 : $r \times n$ 行列 $M =$

, $S \subseteq \{①, ②, \dots, ⓝ\}$

出力 : $\mathrm{span}_M(S)$

$= \{ ⓘ \in \{①, \dots, ⓝ\} \mid \mathrm{rank}(M[S]) = \mathrm{rank}(M[S \cup \{ⓘ\}]) \}$



乱択 $\tilde{O}(\mathrm{nnz}(M) + r^{\omega})$ time

$W := M[S]$ の列空間

① W の直交補空間 $W^{\perp} := \{ v \in \mathbb{F}^r \mid \forall w \in W, v \cdot w = 0 \}$

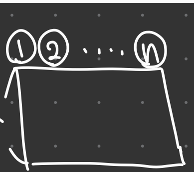
から ランダムなベクトル $\mathbb{r}$ をサンプル

- $W^{\perp}$ の基底 $\{ \mathbb{b}_1, \dots, \mathbb{b}_k \}$ を計算
- $\mathbb{r} = \sum_i r_i \mathbb{b}_i$ (r_i はランダムに選んだ数)

New：高速に $\mathrm{span}_M(S)$ を計算

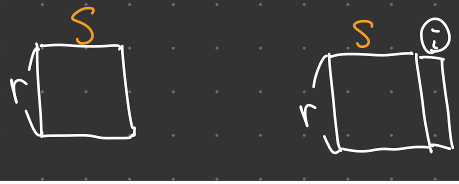
$n \times n$ 行列同士の積：$O(n^\omega)$ $(\omega < 2.372)$

$(r \times n$ 行列$) \times (n \times r$ 行列$)$：$O(nr^{\omega-1})$

入力：$r \times n$ 行列 $M =$  , $S \subseteq \{①, ②, \dots, ⓝ\}$

出力：$\mathrm{span}_M(S)$

$= \{ ⓘ \in \{①, \dots, ⓝ\} \mid \mathrm{rank}(M[S]) = \mathrm{rank}(M[S \cup \{ⓘ\}]) \}$



乱択 $\tilde{O}(\mathrm{nnz}(M) + r^\omega)$ time

$W := M[S]$ の列空間

① W の直交補空間 $W^\perp := \{ v \in \mathbb{F}^r \mid \forall w \in W, v \cdot w = 0 \}$ からランダムなベクトル $\mathbb{r}$ をサンプル

- $W^\perp$ の基底 $\{\mathbb{b}_1, \dots, \mathbb{b}_k\}$ を計算
- $\mathbb{r} = \sum_i r_i \mathbb{b}_i$ (r_i はランダムに選んだ数)

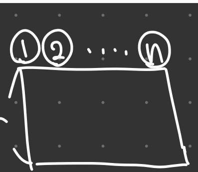
$O(\mathrm{nnz}(M) + r^\omega)$ time

New：高速に $\mathrm{span}_M(S)$ を計算

$n\times n$ 行列同士の積：$O(n^\omega)$ $(\omega<2,372)$

$(r\times n$ 行列$)\times(n\times r$ 行列$)$：$O(nr^{\omega-1})$

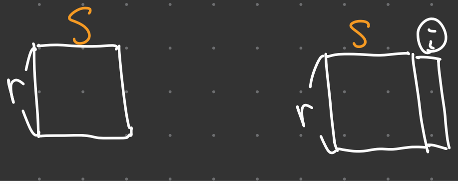
入力：$r\times n$ 行列 $M=$



, $S\subseteq\{①,②,\dots,ⓝ\}$

出力：$\mathrm{span}_M(S)$

$=\{ⓘ\in\{①,\dots,ⓝ\}\mid \mathrm{rank}(M[S])=\mathrm{rank}(M[S\cup\{ⓘ\}])\}$



乱択 $\tilde{O}(\mathrm{nnz}(M)+r^\omega)$ time

$W:=M[S]$ の列空間

① W の直交補空間 $W^\perp:=\{v\in\mathbb{F}^r\mid \forall w\in W,\ v^\top w=0\}$ からランダムなベクトル $\mathbb{r}$ をサンプル

② $\mathrm{span}_M(S)=\{ⓘ\in\{①,\dots,ⓝ\}\mid \mathbb{r}^\top\mathbb{m}_{ⓘ}=0\}$

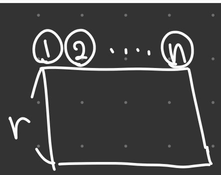
列ⓘに対応する列ベクトル

New：高速に $\mathrm{span}_M(S)$ を計算

$n \times n$ 行列同士の積：$O(n^\omega)$ $(\omega < 2.372)$

$(r \times n$ 行列$) \times (n \times r$ 行列$)$：$O(nr^{\omega-1})$

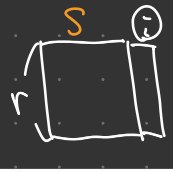
乱択 $\tilde{O}(\mathrm{nnz}(M) + r^\omega)$ time

入力：$r \times n$ 行列 $M =$ 

, $S \subseteq \{①, ②, \dots, ⓝ\}$

出力：$\mathrm{span}_M(S)$

$= \{ⓘ \in \{①, \dots, ⓝ\} \mid \mathrm{rank}(M[S]) = \mathrm{rank}(M[S \cup \{ⓘ\}])\}$



$W := M[S]$ の列空間

① W の直交補空間 $W^\perp := \{v \in \mathbb{F}^r \mid \forall w \in W, v^T w = 0\}$ からランダムなベクトル $\mathbf{r}$ をサンプル

$O(\mathrm{nnz}(M) + r^\omega)$ time

② $\mathrm{span}_M(S) = \{ⓘ \in \{①, \dots, ⓝ\} \mid \mathbf{r}^T \mathbf{m}_ⓘ = 0\}$

列ⓘに対応する列ベクトル

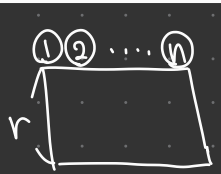
$O(\mathrm{nnz}(M))$ time

New：高速に $\mathrm{span}_M(S)$ を計算

$n\times n$ 行列同士の積：$O(n^\omega)$ $(\omega<2.372)$
$(r\times n$ 行列$)\times(n\times r$ 行列$)$：$O(nr^{\omega-1})$

乱択 $\tilde{O}(\mathrm{nnz}(M)+r^\omega)$ time

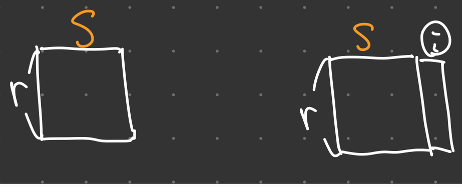
入力：$r\times n$ 行列 $M =$



, $S\subseteq\{$①,②,…,ⓝ$\}$

出力：$\mathrm{span}_M(S)$

$= \{$ⓘ$\in\{$①,…,ⓝ$\}\mid \mathrm{rank}(M[S])=\mathrm{rank}(M[S\cup\{$ⓘ$\}])\}$



$W := M[S]$ の列空間

① W の直交補空間 $W^\perp := \{v\in\mathbb{F}^r \mid \forall w\in W,\ v^\top w=0\}$ からランダムなベクトル $\mathbb{r}$ をサンプル

② $\mathrm{span}_M(S) = \{$ⓘ$\in\{$①,…,ⓝ$\}\mid \mathbb{r}^\top \mathbb{m}_{\text{ⓘ}}=0\}$

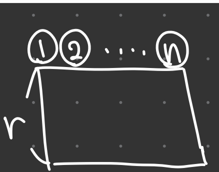
列ⓘに対応する列ベクトル

略証）・ⓘ$\in\mathrm{span}_M(S)$ なら，必ず $\mathbb{r}^\top \mathbb{m}_{\text{ⓘ}}=0$

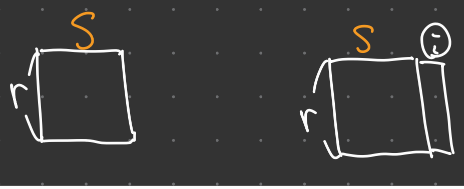
New：高速に $\mathrm{span}_M(S)$ を計算

$n\times n$ 行列同士の積：$O(n^{\omega})$ $(\omega<2.372)$
$(r\times n$ 行列$)\times(n\times r$ 行列$)$：$O(nr^{\omega-1})$

乱択 $\tilde{O}(\mathrm{nnz}(M)+r^{\omega})$ time

入力：$r\times n$ 行列 $M=$ , $S\subseteq\{①,②,\dots,ⓝ\}$

出力：$\mathrm{span}_M(S)$
$=\{ⓘ\in\{①,\dots,ⓝ\}\mid \mathrm{rank}(M[S])=\mathrm{rank}(M[S\cup\{ⓘ\}])\}$



$W:=M[S]$ の列空間

① W の直交補空間 $W^{\perp}:=\{v\in\mathbb{F}^r\mid \forall w\in W,\ v^{T}w=0\}$
からランダムなベクトル $\mathbf{r}$ をサンプル

② $\mathrm{span}_M(S)=\{ⓘ\in\{①,\dots,ⓝ\}\mid \mathbf{r}^{T}\mathbf{m}_{ⓘ}=0\}$

列ⓘに対応する列ベクトル

略証）
- ⓘ $\in\mathrm{span}_M(S)$ なら、必ず $\mathbf{r}^{T}\mathbf{m}_{ⓘ}=0$
- ⓘ $\notin\mathrm{span}_M(S)$ なら、十分高い確率で $\mathbf{r}^{T}\mathbf{m}_{ⓘ}\neq 0$

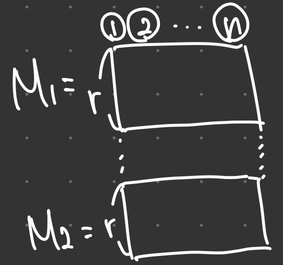
重み付き線形マトロイド交叉

本研究

(1-ε)近似

$\tilde{O}_\varepsilon(\mathrm{nnz}(M_1)+\mathrm{nnz}(M_2)+r_*^{w})$ time

入力：



, 重み $c: \{①,\dots,ⓝ\} \to \mathbb{R}_{\geq 0}$

出力：両方で線形独立な $\sum_{ⓘ \in I} c(ⓘ)$ が最大の列集合 $I \subseteq \{①,\dots,ⓝ\}$

r_* = 重みなしの最適解のサイズ

アイデア

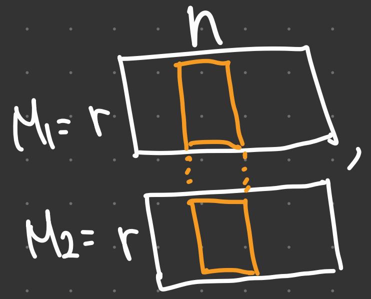
Quanrud'24 + 高速なspan計算アルゴリズム

+ v.d.Brand'21 の高速動的行列公式

+ 内積をランダム重み付き内積に変更

結論

入力:



出力: 両方で線形独立な最大サイズの列集合

$r_* :=$ 最適解のサイズ

サイズ $(1-\varepsilon) r_*$ 以上の解を出力

(重み付き)

- 線形マトロイド交叉問題に対して、

(1-ε)近似 $\tilde{O}_\varepsilon(\mathrm{nnz}(M_1)+\mathrm{nnz}(M_2)+r_*^\omega)$ time

行列の疎さと r_* の小ささを限界まで使った計算量!

アルゴリズムを設計