

Parameterized Complexity of Matrix Factorization

David P. Woodruff

IBM Almaden



Based on joint works with Ilya Razenshteyn, Zhao Song, Peilin Zhong



Outline

- Frobenius norm factorization can be done in polynomial time

Outline

- Frobenius norm factorization can be done in polynomial time
- Intractable matrix problems

Outline

- Frobenius norm factorization can be done in polynomial time
- Intractable matrix problems
 - ▶ Nonnegative matrix factorization(NMF)

Outline

- Frobenius norm factorization can be done in polynomial time
- Intractable matrix problems
 - ▶ Nonnegative matrix factorization(NMF)
 - ★ Arora-Ge-Kannan-Moitra'12, Moitra'13

Outline

- Frobenius norm factorization can be done in polynomial time
- Intractable matrix problems
 - ▶ Nonnegative matrix factorization(NMF)
 - ★ Arora-Ge-Kannan-Moitra'12, Moitra'13
 - ▶ Weighted low rank approximation(WLRA)

Outline

- Frobenius norm factorization can be done in polynomial time
- Intractable matrix problems
 - ▶ Nonnegative matrix factorization(NMF)
 - ★ Arora-Ge-Kannan-Moitra'12, Moitra'13
 - ▶ Weighted low rank approximation(WLRA)
 - ★ Razenshteyn-Song-W'16

Outline

- Frobenius norm factorization can be done in polynomial time
- Intractable matrix problems
 - ▶ Nonnegative matrix factorization(NMF)
 - ★ Arora-Ge-Kannan-Moitra'12, Moitra'13
 - ▶ Weighted low rank approximation(WLRA)
 - ★ Razenshteyn-Song-W'16
 - ▶ ℓ_1 low rank approximation(ℓ_1 LRA)

Outline

- Frobenius norm factorization can be done in polynomial time
- Intractable matrix problems
 - ▶ Nonnegative matrix factorization(NMF)
 - ★ Arora-Ge-Kannan-Moitra'12, Moitra'13
 - ▶ Weighted low rank approximation(WLRA)
 - ★ Razenshteyn-Song-W'16
 - ▶ ℓ_1 low rank approximation(ℓ_1 LRA)
 - ★ Song-W-Zhong'16

Frobenius Norm Matrix Factorization

Question

Given a matrix $A \in \mathbb{R}^{n \times n}$ and $k \geq 1$, the goal is to output two matrices $U \in \mathbb{R}^{n \times k}$, $V \in \mathbb{R}^{k \times n}$ such that

$$\|UV - A\|_F^2 \leq (1 + \epsilon) \min_{\text{rank } k A'} \|A' - A\|_F^2,$$

where $\|A\|_F^2 = (\sum_{i=1}^n \sum_{j=1}^n A_{i,j}^2)^2$, $\epsilon \in [0, 1)$

Frobenius Norm Matrix Factorization

Question

Given a matrix $A \in \mathbb{R}^{n \times n}$ and $k \geq 1$, the goal is to output two matrices $U \in \mathbb{R}^{n \times k}$, $V \in \mathbb{R}^{k \times n}$ such that

$$\|UV - A\|_F^2 \leq (1 + \epsilon) \min_{\text{rank } k A'} \|A' - A\|_F^2,$$

where $\|A\|_F^2 = (\sum_{i=1}^n \sum_{j=1}^n A_{i,j}^2)^2$, $\epsilon \in [0, 1)$

- Useful for data compression, easier to store factorizations U, V

Frobenius Norm Matrix Factorization

Question

Given a matrix $A \in \mathbb{R}^{n \times n}$ and $k \geq 1$, the goal is to output two matrices $U \in \mathbb{R}^{n \times k}$, $V \in \mathbb{R}^{k \times n}$ such that

$$\|UV - A\|_F^2 \leq (1 + \epsilon) \min_{\text{rank } k A'} \|A' - A\|_F^2,$$

where $\|A\|_F^2 = (\sum_{i=1}^n \sum_{j=1}^n A_{i,j}^2)^2$, $\epsilon \in [0, 1)$

- Useful for data compression, easier to store factorizations U, V
- Can be solved using the singular value decomposition (SVD) in matrix multiplication $n^{2.376\dots}$ time

Frobenius Norm Matrix Factorization

Question

Given a matrix $A \in \mathbb{R}^{n \times n}$ and $k \geq 1$, the goal is to output two matrices $U \in \mathbb{R}^{n \times k}$, $V \in \mathbb{R}^{k \times n}$ such that

$$\|UV - A\|_F^2 \leq (1 + \epsilon) \min_{\text{rank } k A'} \|A' - A\|_F^2,$$

where $\|A\|_F^2 = (\sum_{i=1}^n \sum_{j=1}^n A_{i,j}^2)^2$, $\epsilon \in [0, 1)$

- Useful for data compression, easier to store factorizations U, V
- Can be solved using the singular value decomposition (SVD) in matrix multiplication $n^{2.376\dots}$ time
- Fails if you want a nonnegative factorization

Frobenius Norm Matrix Factorization

Question

Given a matrix $A \in \mathbb{R}^{n \times n}$ and $k \geq 1$, the goal is to output two matrices $U \in \mathbb{R}^{n \times k}$, $V \in \mathbb{R}^{k \times n}$ such that

$$\|UV - A\|_F^2 \leq (1 + \epsilon) \min_{\text{rank } k A'} \|A' - A\|_F^2,$$

where $\|A\|_F^2 = (\sum_{i=1}^n \sum_{j=1}^n A_{i,j}^2)^2$, $\epsilon \in [0, 1)$

- Useful for data compression, easier to store factorizations U, V
- Can be solved using the singular value decomposition (SVD) in matrix multiplication $n^{2.376\dots}$ time
- Fails if you want a nonnegative factorization
- Not robust to noise, outliers, or missing entries

Nonnegative Matrix Factorization

Example of NMF

Example of NMF

Given :

$$A \in \mathbb{R}^{n \times n}, n = 4, k = 2$$

Example of NMF

Given :

$$A \in \mathbb{R}^{n \times n}, n = 4, k = 2$$

$$A = \begin{bmatrix} 0 & 1 & 1 & 1 \\ 1 & 2 & 1 & 1 \\ 2 & 4 & 2 & 2 \\ 3 & 5 & 2 & 2 \end{bmatrix}$$

Example of NMF

Given :

$$A \in \mathbb{R}^{n \times n}, n = 4, k = 2$$

Question : Are there two matrices $U, V^\top \in \mathbb{R}^{n \times k}$, such that

$$UV = A, U \geq 0, V \geq 0$$

$$A = \begin{bmatrix} 0 & 1 & 1 & 1 \\ 1 & 2 & 1 & 1 \\ 2 & 4 & 2 & 2 \\ 3 & 5 & 2 & 2 \end{bmatrix}$$

Example of NMF

Given :

$$A \in \mathbb{R}^{n \times n}, n = 4, k = 2$$

Question : Are there two matrices $U, V^\top \in \mathbb{R}^{n \times k}$, such that

$$UV = A, U \geq 0, V \geq 0$$

$$\begin{aligned} A &= \begin{bmatrix} 0 & 1 & 1 & 1 \\ 1 & 2 & 1 & 1 \\ 2 & 4 & 2 & 2 \\ 3 & 5 & 2 & 2 \end{bmatrix} \\ &= \begin{bmatrix} 0 & 1 \\ 1 & 1 \\ 2 & 2 \\ 3 & 2 \end{bmatrix} \cdot \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 \end{bmatrix} \end{aligned}$$

Example of NMF

Given :

$$A \in \mathbb{R}^{n \times n}, n = 4, k = 2$$

Question : Are there two matrices $U, V^\top \in \mathbb{R}^{n \times k}$, such that

$$UV = A, U \geq 0, V \geq 0$$

$$\begin{aligned} A &= \begin{bmatrix} 0 & 1 & 1 & 1 \\ 1 & 2 & 1 & 1 \\ 2 & 4 & 2 & 2 \\ 3 & 5 & 2 & 2 \end{bmatrix} \\ &= \begin{bmatrix} 0 & 1 \\ 1 & 1 \\ 2 & 2 \\ 3 & 2 \end{bmatrix} \cdot \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 \end{bmatrix} \\ &= UV \end{aligned}$$

Main Question, Nonnegative Matrix Factorization

Question

Given matrix $A \in \mathbb{R}^{n \times n}$ and $k \geq 1$, is there an algorithm that can determine if there exist two matrices $U, V^\top \in \mathbb{R}^{n \times k}$,

$$UV = A, U \geq 0, V \geq 0.$$

Or, are there any hardness results?

Main Question, Nonnegative Matrix Factorization

Question

Given matrix $A \in \mathbb{R}^{n \times n}$ and $k \geq 1$, is there an algorithm that can determine if there exist two matrices $U, V^\top \in \mathbb{R}^{n \times k}$,

$$UV = A, U \geq 0, V \geq 0.$$

Or, are there any hardness results?

- Equivalent to computing the nonnegative rank of A , $\text{rank}_+(A)$

Main Question, Nonnegative Matrix Factorization

Question

Given matrix $A \in \mathbb{R}^{n \times n}$ and $k \geq 1$, is there an algorithm that can determine if there exist two matrices $U, V^\top \in \mathbb{R}^{n \times k}$,

$$UV = A, U \geq 0, V \geq 0.$$

Or, are there any hardness results?

- Equivalent to computing the nonnegative rank of A , $\text{rank}_+(A)$
- Fundamental question in machine learning

Main Question, Nonnegative Matrix Factorization

Question

Given matrix $A \in \mathbb{R}^{n \times n}$ and $k \geq 1$, is there an algorithm that can determine if there exist two matrices $U, V^\top \in \mathbb{R}^{n \times k}$,

$$UV = A, U \geq 0, V \geq 0.$$

Or, are there any hardness results?

- Equivalent to computing the nonnegative rank of A , $\text{rank}_+(A)$
- Fundamental question in machine learning
- Applications

Main Question, Nonnegative Matrix Factorization

Question

Given matrix $A \in \mathbb{R}^{n \times n}$ and $k \geq 1$, is there an algorithm that can determine if there exist two matrices $U, V^\top \in \mathbb{R}^{n \times k}$,

$$UV = A, U \geq 0, V \geq 0.$$

Or, are there any hardness results?

- Equivalent to computing the nonnegative rank of A , $\text{rank}_+(A)$
- Fundamental question in machine learning
- Applications
 - ▶ Text mining, Spectral data analysis, Scalable Internet distance prediction, Non-stationary speech denoising, Bioinformatics, Nuclear imaging, etc.

Rank vs. Nonnegative Rank

- rank(A) is the rank and $\text{rank}_+(A)$ is the nonnegative rank

Rank vs. Nonnegative Rank

- $\text{rank}(A)$ is the rank and $\text{rank}_+(A)$ is the nonnegative rank
 - ▶ $\text{rank}(A) \leq \text{rank}_+(A)$

Rank vs. Nonnegative Rank

- $\text{rank}(A)$ is the rank and $\text{rank}_+(A)$ is the nonnegative rank
 - ▶ $\text{rank}(A) \leq \text{rank}_+(A)$
 - ▶ Vavasis'09, determining whether $\text{rank}_+(A) = \text{rank}(A)$ is NP-hard.

Rank vs. Nonnegative Rank

- rank(A) is the rank and $\text{rank}_+(A)$ is the nonnegative rank
 - $\text{rank}(A) \leq \text{rank}_+(A)$
 - Vavasis'09, determining whether $\text{rank}_+(A) = \text{rank}(A)$ is NP-hard.

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

Rank vs. Nonnegative Rank

- rank(A) is the rank and $\text{rank}_+(A)$ is the nonnegative rank
 - rank(A) $\leq \text{rank}_+(A)$
 - Vavasis'09, determining whether $\text{rank}_+(A) = \text{rank}(A)$ is NP-hard.

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

$$\text{rank}(A) = 3,$$

Rank vs. Nonnegative Rank

- rank(A) is the rank and $\text{rank}_+(A)$ is the nonnegative rank
 - $\text{rank}(A) \leq \text{rank}_+(A)$
 - Vavasis'09, determining whether $\text{rank}_+(A) = \text{rank}(A)$ is NP-hard.

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

$\text{rank}(A) = 3$, but $\text{rank}_+(A) = 4$

Polynomial System Verifier

Polynomial System Verifier

[Renegar'92, Basu-Pollack-Roy'96]

Polynomial System Verifier

[Renegar'92, Basu-Pollack-Roy'96]

Given : a polynomial system $P(x)$ over the real numbers

Polynomial System Verifier

[Renegar'92, Basu-Pollack-Roy'96]

Given : a polynomial system $P(x)$ over the real numbers

v : #variables, $x = (x_1, x_2, \dots, x_v)$

Polynomial System Verifier

[Renegar'92, Basu-Pollack-Roy'96]

Given : a polynomial system $P(x)$ over the real numbers

v : #variables, $x = (x_1, x_2, \dots, x_v)$

m : #polynomial constraints $f_i(x) \geq 0, \forall i \in [m]$

Polynomial System Verifier

[Renegar'92, Basu-Pollack-Roy'96]

Given : a polynomial system $P(x)$ over the real numbers

v : #variables, $x = (x_1, x_2, \dots, x_v)$

m : #polynomial constraints $f_i(x) \geq 0, \forall i \in [m]$

d : maximum degree of all polynomial constraints

Polynomial System Verifier

[Renegar'92, Basu-Pollack-Roy'96]

Given : a polynomial system $P(x)$ over the real numbers

v : #variables, $x = (x_1, x_2, \dots, x_v)$

m : #polynomial constraints $f_i(x) \geq 0, \forall i \in [m]$

d : maximum degree of all polynomial constraints

H : the bitsizes of the coefficients of the polynomials

Polynomial System Verifier

[Renegar'92, Basu-Pollack-Roy'96]

Given : a polynomial system $P(x)$ over the real numbers

v : #variables, $x = (x_1, x_2, \dots, x_v)$

m : #polynomial constraints $f_i(x) \geq 0, \forall i \in [m]$

d : maximum degree of all polynomial constraints

H : the bitsizes of the coefficients of the polynomials

In $(md)^{O(v)}$ poly(H) time, can
decide if there exists a solution to polynomial system P

Main Idea

1. Write $\min_{U, V^T \in \mathbb{R}^{n \times k}, U, V \geq 0} \|UV - A\|_F^2$ as a polynomial system
that has $\text{poly}(k)$ variables and $\text{poly}(n)$ constraints and degree
2. Use polynomial system verifier to solve it

Algorithm

Algorithm

Given : $A \in \mathbb{R}^{n \times n}, k \in \mathbb{N}$

Algorithm

Given : $A \in \mathbb{R}^{n \times n}, k \in \mathbb{N}$

Question :

Algorithm

Given : $\textcolor{blue}{A} \in \mathbb{R}^{n \times n}, k \in \mathbb{N}$

Question : Are there matrices $\textcolor{red}{U}, \textcolor{red}{V}^\top \in \mathbb{R}^{n \times k}$ such that

Algorithm

Given : $\textcolor{blue}{A} \in \mathbb{R}^{n \times n}, k \in \mathbb{N}$

Question : Are there matrices $\textcolor{red}{U}, \textcolor{blue}{V}^\top \in \mathbb{R}^{n \times k}$ such that

$$\textcolor{red}{U}\textcolor{blue}{V} = \textcolor{blue}{A}$$

Algorithm

Given : $\textcolor{green}{A} \in \mathbb{R}^{n \times n}, k \in \mathbb{N}$

Question : Are there matrices $\textcolor{red}{U}, \textcolor{red}{V}^\top \in \mathbb{R}^{n \times k}$ such that

$$\textcolor{red}{U}\textcolor{red}{V} = \textcolor{green}{A}$$

$$\textcolor{red}{U} \geqslant 0, \textcolor{red}{V} \geqslant 0$$

Algorithm

Given : $\textcolor{green}{A} \in \mathbb{R}^{n \times n}, k \in \mathbb{N}$

Question : Are there matrices $\textcolor{red}{U}, \textcolor{red}{V}^\top \in \mathbb{R}^{n \times k}$ such that

$$\textcolor{red}{U}\textcolor{red}{V} = \textcolor{green}{A}$$

$$\textcolor{red}{U} \geq 0, \textcolor{red}{V} \geq 0$$

Output :

Algorithm

Given : $\textcolor{green}{A} \in \mathbb{R}^{n \times n}, k \in \mathbb{N}$

Question : Are there matrices $\textcolor{red}{U}, \textcolor{red}{V}^\top \in \mathbb{R}^{n \times k}$ such that

$$\textcolor{red}{U}\textcolor{red}{V} = \textcolor{green}{A}$$

$$\textcolor{red}{U} \geqslant 0, \textcolor{red}{V} \geqslant 0$$

Output : Yes or No

Algorithm

Given : $\textcolor{green}{A} \in \mathbb{R}^{n \times n}, k \in \mathbb{N}$

Question : Are there matrices $\textcolor{red}{U}, \textcolor{red}{V}^\top \in \mathbb{R}^{n \times k}$ such that

$$\textcolor{red}{U}\textcolor{red}{V} = \textcolor{green}{A}$$

$$\textcolor{red}{U} \geq 0, \textcolor{red}{V} \geq 0$$

Output : Yes or No

in $n^{2^{O(k)}}$ time Arora-Ge-Kannan-Moitra'12

Algorithm

Given : $\textcolor{green}{A} \in \mathbb{R}^{n \times n}, k \in \mathbb{N}$

Question : Are there matrices $\textcolor{red}{U}, \textcolor{red}{V}^\top \in \mathbb{R}^{n \times k}$ such that

$$\textcolor{red}{U}\textcolor{red}{V} = \textcolor{green}{A}$$

$$\textcolor{red}{U} \geq 0, \textcolor{red}{V} \geq 0$$

Output : Yes or No

in $n^{2^{O(k)}}$ time Arora-Ge-Kannan-Moitra'12

in $2^{O(k^3)} n^{O(k^2)}$ time Moitra'13

k -SUM

k -SUM

k -SUM

k -SUM

k -SUM

Given: a set of n values $\{s_1, s_2, \dots, s_n\}$ each in the range $[0, 1]$

k -SUM

k -SUM

Given: a set of n values $\{s_1, s_2, \dots, s_n\}$ each in the range $[0, 1]$

Determine:

k -SUM

k -SUM

Given: a set of n values $\{s_1, s_2, \dots, s_n\}$ each in the range $[0, 1]$

Determine: if there is a set of k numbers that sum to exactly $k/2$

k -SUM

k -SUM

Given: a set of n values $\{s_1, s_2, \dots, s_n\}$ each in the range $[0, 1]$

Determine: if there is a set of k numbers that sum to exactly $k/2$

k -SUM hardness [Patrascu-Williams'10]

k -SUM

k -SUM

Given: a set of n values $\{s_1, s_2, \dots, s_n\}$ each in the range $[0, 1]$

Determine: if there is a set of k numbers that sum to exactly $k/2$

k -SUM hardness [Patrascu-Williams'10]

Assume: 3-SAT on n variables cannot be solved in $2^{o(n)}$ time

k -SUM

k -SUM

Given: a set of n values $\{s_1, s_2, \dots, s_n\}$ each in the range $[0, 1]$

Determine: if there is a set of k numbers that sum to exactly $k/2$

k -SUM hardness [Patrascu-Williams'10]

Assume: 3-SAT on n variables cannot be solved in $2^{o(n)}$ time

then k -SUM cannot be solved in $n^{o(k)}$ time

Hardness - Under Exponential Time Hypothesis

Hardness - Under Exponential Time Hypothesis

Given :

$$\textcolor{green}{A} \in \mathbb{R}^{n \times n}, k \in \mathbb{N}$$

Hardness - Under Exponential Time Hypothesis

Given : $A \in \mathbb{R}^{n \times n}$, $k \in \mathbb{N}$

Output : Are there two matrices $U, V^T \in \mathbb{R}^{n \times k}$, such that

$$UV = A$$

$$U \geq 0, V \geq 0$$

Hardness - Under Exponential Time Hypothesis

Given : $A \in \mathbb{R}^{n \times n}$, $k \in \mathbb{N}$

Output : Are there two matrices $U, V^T \in \mathbb{R}^{n \times k}$, such that

$$UV = A$$

$$U \geq 0, V \geq 0$$

Assume : Exponential Time Hypothesis
[Impagliazzo-Paturi-Zane'98]

Hardness - Under Exponential Time Hypothesis

Given : $A \in \mathbb{R}^{n \times n}, k \in \mathbb{N}$

Output : Are there two matrices $U, V^T \in \mathbb{R}^{n \times k}$, such that

$$UV = A$$

$$U \geq 0, V \geq 0$$

Assume : Exponential Time Hypothesis
[Impagliazzo-Paturi-Zane'98]

Requires : $n^{\Omega(k)}$ time

Open Problems

- The upper bound is $n^{O(k^2)}$ while the lower bound is $n^{\Omega(k)}$ - what is the right answer?

Weighted Low Rank Approximation

Weighted Low Rank Approximation

Weighted Low Rank Approximation

Given : $A \in \mathbb{R}^{n \times n}$, $W \in \mathbb{R}^{n \times n}$, $k \in \mathbb{N}$, $\epsilon > 0$

Weighted Low Rank Approximation

Given : $A \in \mathbb{R}^{n \times n}$, $W \in \mathbb{R}^{n \times n}$, $k \in \mathbb{N}$, $\epsilon > 0$

Output :

Weighted Low Rank Approximation

Given : $A \in \mathbb{R}^{n \times n}$, $W \in \mathbb{R}^{n \times n}$, $k \in \mathbb{N}$, $\epsilon > 0$

Output : rank- k $\hat{A} \in \mathbb{R}^{n \times n}$ s.t.

$$\left\| W \circ (\hat{A} - A) \right\|_F^2$$

Weighted Low Rank Approximation

Given : $A \in \mathbb{R}^{n \times n}$, $W \in \mathbb{R}^{n \times n}$, $k \in \mathbb{N}$, $\epsilon > 0$

Output : rank- k $\hat{A} \in \mathbb{R}^{n \times n}$ s.t.

$$\|W \circ (\hat{A} - A)\|_F^2 \leq (1 + \epsilon) \min_{\text{rank-}k A'} \|W \circ (A' - A)\|_F^2$$

$$\left\| \begin{array}{c} W \\ \circ (\quad \hat{A} \quad - \quad A \quad) \end{array} \right\|_F^2$$

Weighted Low Rank Approximation

Given : $A \in \mathbb{R}^{n \times n}$, $W \in \mathbb{R}^{n \times n}$, $k \in \mathbb{N}$, $\epsilon > 0$

Output : rank- k $\hat{A} \in \mathbb{R}^{n \times n}$ s.t.

$$\|W \circ (\hat{A} - A)\|_F^2 \leq (1 + \epsilon) \underbrace{\min_{\text{rank-}k A'} \|W \circ (A' - A)\|_F^2}_{\sum_{i,j} W_{i,j}^2 (A'_{i,j} - A_{i,j})^2}$$

$$\left\| \begin{array}{c} | \\ W \\ | \end{array} \circ \left(\begin{array}{c} | \\ \hat{A} \\ | \end{array} \right) - \begin{array}{c} | \\ A \\ | \end{array} \right\|_F^2$$

Weighted Low Rank Approximation

Given : $A \in \mathbb{R}^{n \times n}$, $W \in \mathbb{R}^{n \times n}$, $k \in \mathbb{N}$, $\epsilon > 0$

Output : rank- k $\hat{A} \in \mathbb{R}^{n \times n}$ s.t.

$$\|W \circ (\hat{A} - A)\|_F^2 \leq (1 + \epsilon) \underbrace{\min_{\text{rank-}k A'} \|W \circ (A' - A)\|_F^2}_{\text{OPT}} + \underbrace{\sum_{i,j} W_{i,j}^2 (A'_{i,j} - A_{i,j})^2}_{\text{OPT}}$$

$$\left\| \begin{array}{c} | \\ W \\ | \end{array} \circ \left(\begin{array}{c} | \\ \hat{A} \\ | \end{array} \right) - \begin{array}{c} | \\ A \\ | \end{array} \right\|_F^2$$

Weighted Low Rank Approximation

Given : $A \in \mathbb{R}^{n \times n}$, $W \in \mathbb{R}^{n \times n}$, $k \in \mathbb{N}$, $\epsilon > 0$

Output : $U, V^\top \in \mathbb{R}^{n \times k}$ s.t.
 $\|W \circ (UV - A)\|_F^2 \leq (1 + \epsilon) \text{OPT}$

$$\left\| \begin{array}{c} W \\ \circ (\begin{array}{c} U \\ V \end{array}) - \begin{array}{c} A \end{array} \end{array} \right\|_F^2$$

Motivation

Motivation



Motivation



Motivation



Action

Comedy

Historical

Cartoon

Magical



Motivation



Action

Comedy

Historical

Cartoon

Magical



9 8

Motivation



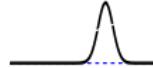
Action

Comedy

Historical

Cartoon

Magical



Motivation



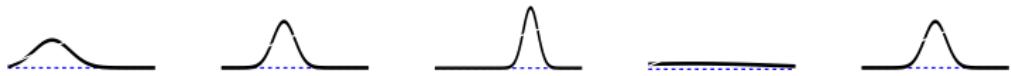
Action

Comedy

Historical

Cartoon

Magical



Motivation



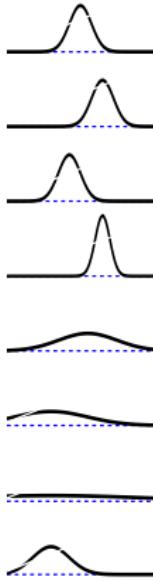
Action

Comedy

Historical

Cartoon

Magical



Motivation



Action

Comedy

Historical

Cartoon

Magical



$\frac{1}{5}$ $\frac{1}{5}$



$\frac{1}{1}$ $\frac{1}{1}$



$\frac{1}{1}$ $\frac{1}{1}$



$\frac{1}{5}$ $\frac{1}{5}$



$\frac{1}{1}$ $\frac{1}{1}$



$\frac{1}{4}$ $\frac{1}{4}$



$\frac{1}{6}$ $\frac{1}{6}$



$\frac{1}{1}$ $\frac{1}{1}$

Motivation



Action



$$\frac{1}{5} \quad \frac{1}{5}$$

Comedy



$$\frac{1}{1} \quad \frac{1}{1}$$

Historical



$$\frac{1}{1} \quad \frac{1}{1}$$

Cartoon



$$\frac{1}{5} \quad \frac{1}{5}$$

Magical



$$\frac{1}{1} \quad \frac{1}{1}$$



$$\frac{1}{1} \quad \frac{1}{3}$$



$$\frac{1}{4} \quad \frac{1}{4}$$



$$\frac{1}{6} \quad \frac{1}{6}$$



$$\frac{1}{1} \quad \frac{1}{1}$$

Motivation



Action

Comedy

Historical

Cartoon

Magical



$\frac{1}{5}$



$\frac{1}{1}$



$\frac{1}{1}$



$\frac{1}{5}$



$\frac{1}{1}$



$\frac{1}{4}$



$\frac{1}{6}$



$\frac{1}{1}$

$\frac{1}{1}$

$\frac{1}{1}$

$\frac{1}{1}$

$\frac{1}{6}$

$\frac{1}{3}$

$\frac{1}{2}$

$\frac{1}{2}$

$\frac{1}{2}$

$\frac{1}{1}$

$\frac{1}{1}$

$\frac{1}{1}$

$\frac{1}{6}$

$\frac{1}{3}$

$\frac{1}{2}$

$\frac{1}{2}$

$\frac{1}{4}$

$\frac{1}{3}$

$\frac{1}{6}$

$\frac{1}{4}$

$\frac{1}{2}$

$\frac{1}{7}$

$\frac{1}{5}$

$\frac{1}{4}$

$\frac{1}{1}$

Motivation



Action

Comedy

Historical

Cartoon

Magical


 $\frac{1}{5}$
 $\frac{1}{1}$
 $\frac{1}{1}$
 $\frac{1}{5}$
 $\frac{1}{1}$
 $\frac{1}{3}$
 $\frac{1}{4}$
 $\frac{1}{6}$
 $\frac{1}{1}$
 $\frac{1}{1}$
 $\frac{1}{1}$
 $\frac{1}{1}$
 $\frac{1}{6}$
 $\frac{1}{3}$
 $\frac{1}{2}$
 $\frac{1}{2}$
 $\frac{1}{2}$
 $\frac{1}{4}$
 $\frac{1}{4}$
 $\frac{1}{3}$
 $\frac{1}{6}$
 $\frac{1}{4}$
 $\frac{1}{2}$
 $\frac{1}{7}$
 $\frac{1}{5}$
 $\frac{1}{5}$
 $\frac{1}{4}$
 $\frac{1}{4}$
 $\frac{1}{1}$
 $\frac{1}{2}$
 $\frac{1}{1}$
 $\frac{1}{2}$
 $\frac{1}{1}$
 $\frac{1}{5}$
 $\frac{1}{1}$
 $\frac{1}{3}$
 $\frac{1}{4}$
 $\frac{1}{3}$
 $\frac{1}{4}$
 $\frac{1}{2}$
 $\frac{1}{1}$
 $\frac{1}{2}$
 $\frac{1}{1}$
 $\frac{1}{5}$
 $\frac{1}{1}$
 $\frac{1}{3}$
 $\frac{1}{4}$
 $\frac{1}{3}$
 $\frac{1}{4}$

Motivation



Action

Comedy

Historical

Cartoon

Magical


 $\frac{1}{5}$
 $\frac{1}{5}$
 $\frac{1}{1}$
 $\frac{1}{5}$
 $\frac{1}{1}$
 $\frac{1}{3}$
 $\frac{1}{4}$
 $\frac{1}{6}$
 $\frac{1}{1}$
 $\frac{1}{1}$
 $\frac{1}{1}$
 $\frac{1}{1}$
 $\frac{1}{6}$
 $\frac{1}{3}$
 $\frac{1}{2}$
 $\frac{1}{2}$
 $\frac{1}{4}$
 $\frac{1}{2}$
 $\frac{1}{3}$
 $\frac{1}{6}$
 $\frac{1}{4}$
 $\frac{1}{2}$
 $\frac{1}{7}$
 $\frac{1}{5}$
 $\frac{1}{5}$
 $\frac{1}{4}$
 $\frac{1}{1}$
 $\frac{1}{2}$
 $\frac{1}{1}$
 $\frac{1}{2}$
 $\frac{1}{1}$
 $\frac{1}{5}$
 $\frac{1}{1}$
 $\frac{1}{1}$
 $\frac{1}{3}$
 $\frac{1}{4}$
 $\frac{1}{2}$
 $\frac{1}{2}$
 $\frac{1}{6}$
 $\frac{1}{4}$
 $\frac{1}{2}$
 $\frac{1}{1}$
 $\frac{1}{3}$
 $\frac{1}{6}$
 $\frac{1}{2}$
 $\frac{1}{2}$
 $\frac{1}{2}$
 $\frac{1}{6}$
 $\frac{1}{4}$
 $\frac{1}{2}$
 $\frac{1}{1}$
 $\frac{1}{3}$
 $\frac{1}{6}$
 $\frac{1}{2}$

Motivation



Action

Comedy

Historical

Cartoon

Magical



Motivation



8 8

8 8



Motivation



Action

Comedy

Historical

Cartoon

Magical



8 8

8 8

4 4

4 4



Motivation



Action

Comedy

Historical

Cartoon

Magical



8 8



8 8



4 4



4 4

1 1

1 1

Motivation



Action

Comedy

Historical

Cartoon

Magical



8 8

8 8



4 4

4 4



1 1

1 1



0 0

0 0

Motivation



Action

Comedy

Historical

Cartoon

Magical



8 8

8 8



4 4



4 4

1 1

1 1



0 0 8 8

0 0 8 8

Motivation



Action

Comedy

Historical

Cartoon

Magical



8 8

8 8



4 4

4 4



1 1 4 4

1 1 4 4



0 0 8 8

0 0 8 8

Motivation



Action

Comedy

Historical

Cartoon

Magical



8 8

8 8



4 4 2 2



4 4 2 2



1 1 4 4



1 1 4 4

0 0 8 8

0 0 8 8

Motivation



Action

Comedy

Historical

Cartoon

Magical



8 8 1 1

8 8 1 1



4 4 2 2



4 4 2 2

1 1 4 4

1 1 4 4



0 0 8 8

0 0 8 8

Matrix Completion

Matrix Completion

Given : $A \in \{\mathbb{R}, ?\}^{n \times n}$, $\Omega \subset [n] \times [n]$ and $k \in \mathbb{R}$

Matrix Completion

Given : $A \in \{\mathbb{R}, ?\}^{n \times n}$, $\Omega \subset [n] \times [n]$ and $k \in \mathbb{R}$

$$A = \begin{bmatrix} 1 & ? & ? & 0 \\ ? & 1 & 0 & ? \\ ? & 0 & 1 & ? \\ 0 & ? & ? & 1 \end{bmatrix}, k = 2$$

Matrix Completion

Given : $A \in \{\mathbb{R}, ?\}^{n \times n}$, $\Omega \subset [n] \times [n]$ and $k \in \mathbb{R}$

$$A = \begin{bmatrix} 1 & ? & ? & 0 \\ ? & 1 & 0 & ? \\ ? & 0 & 1 & ? \\ 0 & ? & ? & 1 \end{bmatrix}, k = 2$$

Output : $A_{\bar{\Omega}}$ s.t. $\text{rank}(A) = k$

Matrix Completion

Given : $A \in \{\mathbb{R}, ?\}^{n \times n}$, $\Omega \subset [n] \times [n]$ and $k \in \mathbb{R}$

$$A = \begin{bmatrix} 1 & ? & ? & 0 \\ ? & 1 & 0 & ? \\ ? & 0 & 1 & ? \\ 0 & ? & ? & 1 \end{bmatrix}, k = 2$$

Output : $A_{\bar{\Omega}}$ s.t. $\text{rank}(A) = k$

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

Matrix Completion

Given : $A \in \{\mathbb{R}, ?\}^{n \times n}$, $\Omega \subset [n] \times [n]$ and $k \in \mathbb{R}$

$$A = \begin{bmatrix} 1 & ? & ? & 0 \\ ? & 1 & 0 & ? \\ ? & 0 & 1 & ? \\ 0 & ? & ? & 1 \end{bmatrix}, k = 2$$

Output : $A_{\bar{\Omega}}$ s.t. $\text{rank}(A) = k$

Algorithms : Candes-Recht'09, Candes-Recht'10, Keshavan'12
Hardt-Wootters'14, Jain-Netrapalli-Sanghavi'13
Hardt'15, Sun-Luo'15

Matrix Completion

Given : $A \in \{\mathbb{R}, ?\}^{n \times n}$, $\Omega \subset [n] \times [n]$ and $k \in \mathbb{R}$

$$A = \begin{bmatrix} 1 & ? & ? & 0 \\ ? & 1 & 0 & ? \\ ? & 0 & 1 & ? \\ 0 & ? & ? & 1 \end{bmatrix}, k = 2$$

Output : $A_{\bar{\Omega}}$ s.t. $\text{rank}(A) = k$

Algorithms : Candes-Recht'09, Candes-Recht'10, Keshavan'12

Hardt-Wootters'14, Jain-Netrapalli-Sanghavi'13

Hardt'15, Sun-Luo'15

Hardness : Peeters'96, Gillis-Glineur'11

Hardt-Meka-Raghavendra-Weitz'14

More Real-life Datasets

More Real-life Datasets

RateBeer



Tasters



Beers



Ratings

More Real-life Datasets

RateBeer



Documents



Words



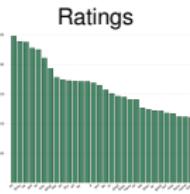
Beers



★ 533 reviews



Books



Frequency

More Real-life Datasets

RateBeer



Tasters



Documents



Beers



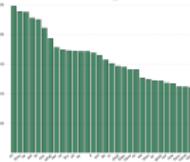
Amazon



Users



Ratings



Frequency

Customer Reviews

1,320
4.1 out of 5 stars



[See all 1,320 customer reviews](#)

Books



Products

Ratings

More Real-life Datasets

RateBeer



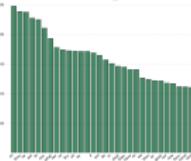
Tasters



Beers



Ratings



Documents

Amazon

Words



Books



Frequency

Customer Reviews

★ ★ ★ ★ ★ 1,320
4.1 out of 5 stars



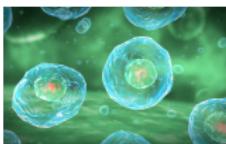
Biology

Users

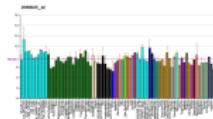


Genes

Products



Ratings



Main Results - Summary

- Algorithm for Weighted low rank approximation(WLRA) problem

Main Results - Summary

- Algorithm for Weighted low rank approximation(WLRA) problem
 - W has r distinct rows and columns

$$W = \begin{bmatrix} 1 & 1 & 1 & 3 & 3 & 4 \\ 1 & 1 & 1 & 3 & 3 & 4 \\ 2 & 2 & 2 & 5 & 5 & 0 \\ 2 & 2 & 2 & 5 & 5 & 0 \\ 2 & 2 & 2 & 5 & 5 & 0 \\ 7 & 7 & 7 & 4 & 4 & 6 \end{bmatrix}, r = 3$$

Main Results - Summary

- Algorithm for Weighted low rank approximation(WLRA) problem
 - W has r distinct rows and columns
 - W has r distinct columns

$$W = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ 2 & 2 & 2 & 1 & 1 & 3 \\ 3 & 3 & 3 & 1 & 1 & 5 \\ 4 & 4 & 4 & 1 & 1 & 1 \\ 5 & 5 & 5 & 1 & 1 & 3 \\ 6 & 6 & 6 & 1 & 1 & 5 \end{bmatrix}, r = 3$$

Main Results - Summary

- Algorithm for Weighted low rank approximation(WLRA) problem
 - W has r distinct rows and columns
 - W has r distinct columns
 - W has rank at most r

$$W = \begin{bmatrix} 1 & 1 & 1 & 2 & 2 & 2 \\ 0 & 1 & 2 & 1 & 2 & 3 \\ 1 & 1 & 3 & 2 & 4 & 4 \\ 0 & 0 & 4 & 0 & 4 & 4 \\ 1 & 0 & 5 & 1 & 6 & 5 \\ 0 & 0 & 6 & 0 & 6 & 6 \end{bmatrix}, r = 3$$

Main Results - Summary

- Algorithm for Weighted low rank approximation(WLRA) problem
 - W has r distinct rows and columns
 - W has r distinct columns
 - W has rank at most r
- Hardness for Weighted low rank approximation(WLRA) problem

r Distinct Rows and Columns

r Distinct Rows and Columns

Given : $A \in \mathbb{R}^{n \times n}$, $W \in \mathbb{R}^{n \times n}$, $r \in \mathbb{N}$, $k \in \mathbb{N}$, $\epsilon > 0$

r Distinct Rows and Columns

Given : $A \in \mathbb{R}^{n \times n}$, $W \in \mathbb{R}^{n \times n}$, $r \in \mathbb{N}$, $k \in \mathbb{N}$, $\epsilon > 0$

$$W = \begin{bmatrix} 1 & 1 & 1 & 3 & 3 & 4 \\ 1 & 1 & 1 & 3 & 3 & 4 \\ 2 & 2 & 2 & 5 & 5 & 0 \\ 2 & 2 & 2 & 5 & 5 & 0 \\ 2 & 2 & 2 & 5 & 5 & 0 \\ 7 & 7 & 7 & 4 & 4 & 6 \end{bmatrix}, r = 3$$

r Distinct Rows and Columns

Given : $A \in \mathbb{R}^{n \times n}$, $W \in \mathbb{R}^{n \times n}$, $r \in \mathbb{N}$, $k \in \mathbb{N}$, $\epsilon > 0$

$$W = \begin{bmatrix} 1 & 1 & 1 & 3 & 3 & 4 \\ 1 & 1 & 1 & 3 & 3 & 4 \\ 2 & 2 & 2 & 5 & 5 & 0 \\ 2 & 2 & 2 & 5 & 5 & 0 \\ 2 & 2 & 2 & 5 & 5 & 0 \\ 7 & 7 & 7 & 4 & 4 & 6 \end{bmatrix}, r = 3$$

Output :

r Distinct Rows and Columns

Given : $A \in \mathbb{R}^{n \times n}$, $W \in \mathbb{R}^{n \times n}$, $r \in \mathbb{N}$, $k \in \mathbb{N}$, $\epsilon > 0$

$$W = \begin{bmatrix} 1 & 1 & 1 & 3 & 3 & 4 \\ 1 & 1 & 1 & 3 & 3 & 4 \\ 2 & 2 & 2 & 5 & 5 & 0 \\ 2 & 2 & 2 & 5 & 5 & 0 \\ 2 & 2 & 2 & 5 & 5 & 0 \\ 7 & 7 & 7 & 4 & 4 & 6 \end{bmatrix}, r = 3$$

Output : rank- k $\hat{A} \in \mathbb{R}^{n \times n}$ s.t.

r Distinct Rows and Columns

Given : $A \in \mathbb{R}^{n \times n}$, $W \in \mathbb{R}^{n \times n}$, $r \in \mathbb{N}$, $k \in \mathbb{N}$, $\epsilon > 0$

$$W = \begin{bmatrix} 1 & 1 & 1 & 3 & 3 & 4 \\ 1 & 1 & 1 & 3 & 3 & 4 \\ 2 & 2 & 2 & 5 & 5 & 0 \\ 2 & 2 & 2 & 5 & 5 & 0 \\ 2 & 2 & 2 & 5 & 5 & 0 \\ 7 & 7 & 7 & 4 & 4 & 6 \end{bmatrix}, r = 3$$

Output : rank- k $\hat{A} \in \mathbb{R}^{n \times n}$ s.t.

$$\|W \circ (A - \hat{A})\|_F^2 \leq (1 + \epsilon) \min_{\text{rank-}k A'} \|W \circ (A - A')\|_F^2$$

r Distinct Rows and Columns

Given : $A \in \mathbb{R}^{n \times n}$, $W \in \mathbb{R}^{n \times n}$, $r \in \mathbb{N}$, $k \in \mathbb{N}$, $\epsilon > 0$

$$W = \begin{bmatrix} 1 & 1 & 1 & 3 & 3 & 4 \\ 1 & 1 & 1 & 3 & 3 & 4 \\ 2 & 2 & 2 & 5 & 5 & 0 \\ 2 & 2 & 2 & 5 & 5 & 0 \\ 2 & 2 & 2 & 5 & 5 & 0 \\ 7 & 7 & 7 & 4 & 4 & 6 \end{bmatrix}, r = 3$$

Output : rank- k $\hat{A} \in \mathbb{R}^{n \times n}$ s.t.

$$\|W \circ (A - \hat{A})\|_F^2 \leq (1 + \epsilon) \min_{\text{rank-}k A'} \sum_{i,j} W_{i,j}^2 (A_{i,j} - A'_{i,j})^2$$

r Distinct Rows and Columns

Given : $A \in \mathbb{R}^{n \times n}$, $W \in \mathbb{R}^{n \times n}$, $r \in \mathbb{N}$, $k \in \mathbb{N}$, $\epsilon > 0$

$$W = \begin{bmatrix} 1 & 1 & 1 & 3 & 3 & 4 \\ 1 & 1 & 1 & 3 & 3 & 4 \\ 2 & 2 & 2 & 5 & 5 & 0 \\ 2 & 2 & 2 & 5 & 5 & 0 \\ 2 & 2 & 2 & 5 & 5 & 0 \\ 7 & 7 & 7 & 4 & 4 & 6 \end{bmatrix}, r = 3$$

Output : rank- k $\hat{A} \in \mathbb{R}^{n \times n}$ s.t.

$$\|W \circ (A - \hat{A})\|_F^2 \leq (1 + \epsilon) \text{OPT}$$

r Distinct Rows and Columns

Given : $A \in \mathbb{R}^{n \times n}$, $W \in \mathbb{R}^{n \times n}$, $r \in \mathbb{N}$, $k \in \mathbb{N}$, $\epsilon > 0$

$$W = \begin{bmatrix} 1 & 1 & 1 & 3 & 3 & 4 \\ 1 & 1 & 1 & 3 & 3 & 4 \\ 2 & 2 & 2 & 5 & 5 & 0 \\ 2 & 2 & 2 & 5 & 5 & 0 \\ 2 & 2 & 2 & 5 & 5 & 0 \\ 7 & 7 & 7 & 4 & 4 & 6 \end{bmatrix}, r = 3$$

Output : rank- k $\hat{A} \in \mathbb{R}^{n \times n}$ s.t.

$$\|W \circ (A - \hat{A})\|_F^2 \leq (1 + \epsilon) \text{OPT}$$

with prob. 9/10

r Distinct Rows and Columns

Given : $A \in \mathbb{R}^{n \times n}$, $W \in \mathbb{R}^{n \times n}$, $r \in \mathbb{N}$, $k \in \mathbb{N}$, $\epsilon > 0$

$$W = \begin{bmatrix} 1 & 1 & 1 & 3 & 3 & 4 \\ 1 & 1 & 1 & 3 & 3 & 4 \\ 2 & 2 & 2 & 5 & 5 & 0 \\ 2 & 2 & 2 & 5 & 5 & 0 \\ 2 & 2 & 2 & 5 & 5 & 0 \\ 7 & 7 & 7 & 4 & 4 & 6 \end{bmatrix}, r = 3$$

Output : rank- k $\hat{A} \in \mathbb{R}^{n \times n}$ s.t.

$$\|W \circ (A - \hat{A})\|_F^2 \leq (1 + \epsilon) \text{OPT}$$

with prob. 9/10

in $O((\text{nnz}(A) + \text{nnz}(W)) \cdot n^\gamma) + n \cdot 2^{\tilde{O}(k^2r/\epsilon)}$ time

r Distinct Rows and Columns

Given : $A \in \mathbb{R}^{n \times n}$, $W \in \mathbb{R}^{n \times n}$, $r \in \mathbb{N}$, $k \in \mathbb{N}$, $\epsilon > 0$

$$W = \begin{bmatrix} 1 & 1 & 1 & 3 & 3 & 4 \\ 1 & 1 & 1 & 3 & 3 & 4 \\ 2 & 2 & 2 & 5 & 5 & 0 \\ 2 & 2 & 2 & 5 & 5 & 0 \\ 2 & 2 & 2 & 5 & 5 & 0 \\ 7 & 7 & 7 & 4 & 4 & 6 \end{bmatrix}, r = 3$$

Output : rank- k $\hat{A} \in \mathbb{R}^{n \times n}$ s.t.

$$\|W \circ (A - \hat{A})\|_F^2 \leq (1 + \epsilon) \text{OPT}$$

with prob. 9/10

in $O((\text{nnz}(A) + \text{nnz}(W)) \cdot n^\gamma) + n \cdot 2^{\tilde{O}(k^2r/\epsilon)}$ time
for an arbitrarily small constant $\gamma > 0$

r Distinct Rows and Columns

Given : $A \in \mathbb{R}^{n \times n}$, $W \in \mathbb{R}^{n \times n}$, $r \in \mathbb{N}$, $k \in \mathbb{N}$, $\epsilon > 0$

$$W = \begin{bmatrix} 1 & 1 & 1 & 3 & 3 & 4 \\ 1 & 1 & 1 & 3 & 3 & 4 \\ 2 & 2 & 2 & 5 & 5 & 0 \\ 2 & 2 & 2 & 5 & 5 & 0 \\ 2 & 2 & 2 & 5 & 5 & 0 \\ 7 & 7 & 7 & 4 & 4 & 6 \end{bmatrix}, r = 3$$

Output : rank- k $\hat{A} \in \mathbb{R}^{n \times n}$ s.t.

$$\|W \circ (A - \hat{A})\|_F^2 \leq (1 + \epsilon) \text{OPT}$$

with prob. 9/10

in $O((\text{nnz}(A) + \text{nnz}(W)) \cdot n^\gamma) + n \cdot 2^{\tilde{O}(k^2r/\epsilon)}$ time
for an arbitrarily small constant $\gamma > 0$

fixed parameter tractable

r Distinct Columns

r Distinct Columns

Given : $A \in \mathbb{R}^{n \times n}$, $W \in \mathbb{R}^{n \times n}$, $r \in \mathbb{N}$, $k \in \mathbb{N}$, $\epsilon > 0$

r Distinct Columns

Given : $A \in \mathbb{R}^{n \times n}$, $W \in \mathbb{R}^{n \times n}$, $r \in \mathbb{N}$, $k \in \mathbb{N}$, $\epsilon > 0$

$$W = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ 2 & 2 & 2 & 1 & 1 & 3 \\ 3 & 3 & 3 & 1 & 1 & 5 \\ 4 & 4 & 4 & 1 & 1 & 1 \\ 5 & 5 & 5 & 1 & 1 & 3 \\ 6 & 6 & 6 & 1 & 1 & 5 \end{bmatrix}, r = 3$$

r Distinct Columns

Given : $A \in \mathbb{R}^{n \times n}$, $W \in \mathbb{R}^{n \times n}$, $r \in \mathbb{N}$, $k \in \mathbb{N}$, $\epsilon > 0$

$$W = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ 2 & 2 & 2 & 1 & 1 & 3 \\ 3 & 3 & 3 & 1 & 1 & 5 \\ 4 & 4 & 4 & 1 & 1 & 1 \\ 5 & 5 & 5 & 1 & 1 & 3 \\ 6 & 6 & 6 & 1 & 1 & 5 \end{bmatrix}, r = 3$$

Output :

r Distinct Columns

Given : $A \in \mathbb{R}^{n \times n}$, $W \in \mathbb{R}^{n \times n}$, $r \in \mathbb{N}$, $k \in \mathbb{N}$, $\epsilon > 0$

$$W = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ 2 & 2 & 2 & 1 & 1 & 3 \\ 3 & 3 & 3 & 1 & 1 & 5 \\ 4 & 4 & 4 & 1 & 1 & 1 \\ 5 & 5 & 5 & 1 & 1 & 3 \\ 6 & 6 & 6 & 1 & 1 & 5 \end{bmatrix}, r = 3$$

Output : rank- k $\hat{A} \in \mathbb{R}^{n \times n}$ s.t.

r Distinct Columns

Given : $A \in \mathbb{R}^{n \times n}$, $W \in \mathbb{R}^{n \times n}$, $r \in \mathbb{N}$, $k \in \mathbb{N}$, $\epsilon > 0$

$$W = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ 2 & 2 & 2 & 1 & 1 & 3 \\ 3 & 3 & 3 & 1 & 1 & 5 \\ 4 & 4 & 4 & 1 & 1 & 1 \\ 5 & 5 & 5 & 1 & 1 & 3 \\ 6 & 6 & 6 & 1 & 1 & 5 \end{bmatrix}, r = 3$$

Output : rank- k $\hat{A} \in \mathbb{R}^{n \times n}$ s.t.

$$\|W \circ (A - \hat{A})\|_F^2 \leq (1 + \epsilon) \min_{\text{rank-}k A'} \|W \circ (A - A')\|_F^2$$

r Distinct Columns

Given : $A \in \mathbb{R}^{n \times n}$, $W \in \mathbb{R}^{n \times n}$, $r \in \mathbb{N}$, $k \in \mathbb{N}$, $\epsilon > 0$

$$W = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ 2 & 2 & 2 & 1 & 1 & 3 \\ 3 & 3 & 3 & 1 & 1 & 5 \\ 4 & 4 & 4 & 1 & 1 & 1 \\ 5 & 5 & 5 & 1 & 1 & 3 \\ 6 & 6 & 6 & 1 & 1 & 5 \end{bmatrix}, r = 3$$

Output : rank- k $\hat{A} \in \mathbb{R}^{n \times n}$ s.t.

$$\|W \circ (A - \hat{A})\|_F^2 \leq (1 + \epsilon) \min_{\text{rank-}k A'} \sum_{i,j} W_{i,j}^2 (A_{i,j} - A'_{i,j})^2$$

r Distinct Columns

Given : $A \in \mathbb{R}^{n \times n}$, $W \in \mathbb{R}^{n \times n}$, $r \in \mathbb{N}$, $k \in \mathbb{N}$, $\epsilon > 0$

$$W = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ 2 & 2 & 2 & 1 & 1 & 3 \\ 3 & 3 & 3 & 1 & 1 & 5 \\ 4 & 4 & 4 & 1 & 1 & 1 \\ 5 & 5 & 5 & 1 & 1 & 3 \\ 6 & 6 & 6 & 1 & 1 & 5 \end{bmatrix}, r = 3$$

Output : rank- k $\hat{A} \in \mathbb{R}^{n \times n}$ s.t.

$$\|W \circ (A - \hat{A})\|_F^2 \leq (1 + \epsilon) \text{OPT}$$

r Distinct Columns

Given : $A \in \mathbb{R}^{n \times n}$, $W \in \mathbb{R}^{n \times n}$, $r \in \mathbb{N}$, $k \in \mathbb{N}$, $\epsilon > 0$

$$W = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ 2 & 2 & 2 & 1 & 1 & 3 \\ 3 & 3 & 3 & 1 & 1 & 5 \\ 4 & 4 & 4 & 1 & 1 & 1 \\ 5 & 5 & 5 & 1 & 1 & 3 \\ 6 & 6 & 6 & 1 & 1 & 5 \end{bmatrix}, r = 3$$

Output : rank- k $\hat{A} \in \mathbb{R}^{n \times n}$ s.t.

$$\|W \circ (A - \hat{A})\|_F^2 \leq (1 + \epsilon) \text{OPT}$$

with prob. 9/10

r Distinct Columns

Given : $A \in \mathbb{R}^{n \times n}$, $W \in \mathbb{R}^{n \times n}$, $r \in \mathbb{N}$, $k \in \mathbb{N}$, $\epsilon > 0$

$$W = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ 2 & 2 & 2 & 1 & 1 & 3 \\ 3 & 3 & 3 & 1 & 1 & 5 \\ 4 & 4 & 4 & 1 & 1 & 1 \\ 5 & 5 & 5 & 1 & 1 & 3 \\ 6 & 6 & 6 & 1 & 1 & 5 \end{bmatrix}, r = 3$$

Output : rank- k $\hat{A} \in \mathbb{R}^{n \times n}$ s.t.

$$\|W \circ (A - \hat{A})\|_F^2 \leq (1 + \epsilon) \text{OPT}$$

with prob. 9/10

in $O((\text{nnz}(A) + \text{nnz}(W)) \cdot n^\gamma) + n \cdot 2^{\tilde{O}(k^2 r^2 / \epsilon)}$ time

r Distinct Columns

Given : $A \in \mathbb{R}^{n \times n}$, $W \in \mathbb{R}^{n \times n}$, $r \in \mathbb{N}$, $k \in \mathbb{N}$, $\epsilon > 0$

$$W = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ 2 & 2 & 2 & 1 & 1 & 3 \\ 3 & 3 & 3 & 1 & 1 & 5 \\ 4 & 4 & 4 & 1 & 1 & 1 \\ 5 & 5 & 5 & 1 & 1 & 3 \\ 6 & 6 & 6 & 1 & 1 & 5 \end{bmatrix}, r = 3$$

Output : rank- k $\hat{A} \in \mathbb{R}^{n \times n}$ s.t.

$$\|W \circ (A - \hat{A})\|_F^2 \leq (1 + \epsilon) \text{OPT}$$

with prob. 9/10

in $O((\text{nnz}(A) + \text{nnz}(W)) \cdot n^\gamma) + n \cdot 2^{\tilde{O}(k^2 r^2 / \epsilon)}$ time
for an arbitrarily small constant $\gamma > 0$

r Distinct Columns

Given : $A \in \mathbb{R}^{n \times n}$, $W \in \mathbb{R}^{n \times n}$, $r \in \mathbb{N}$, $k \in \mathbb{N}$, $\epsilon > 0$

$$W = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ 2 & 2 & 2 & 1 & 1 & 3 \\ 3 & 3 & 3 & 1 & 1 & 5 \\ 4 & 4 & 4 & 1 & 1 & 1 \\ 5 & 5 & 5 & 1 & 1 & 3 \\ 6 & 6 & 6 & 1 & 1 & 5 \end{bmatrix}, r = 3$$

Output : rank- k $\hat{A} \in \mathbb{R}^{n \times n}$ s.t.

$$\|W \circ (A - \hat{A})\|_F^2 \leq (1 + \epsilon) \text{OPT}$$

with prob. 9/10

in $O((\text{nnz}(A) + \text{nnz}(W)) \cdot n^\gamma) + n \cdot 2^{\tilde{O}(k^2 r^2 / \epsilon)}$ time
for an arbitrarily small constant $\gamma > 0$

fixed parameter tractable

Rank r

Rank r

Given : $A \in \mathbb{R}^{n \times n}$, $W \in \mathbb{R}^{n \times n}$, $r \in \mathbb{N}$, $k \in \mathbb{N}$, $\epsilon > 0$

Rank r

Given : $A \in \mathbb{R}^{n \times n}$, $W \in \mathbb{R}^{n \times n}$, $r \in \mathbb{N}$, $k \in \mathbb{N}$, $\epsilon > 0$

$$W = \begin{bmatrix} 1 & 1 & 1 & 2 & 2 & 2 \\ 0 & 1 & 2 & 1 & 2 & 3 \\ 1 & 1 & 3 & 2 & 4 & 4 \\ 0 & 0 & 4 & 0 & 4 & 4 \\ 1 & 0 & 5 & 1 & 6 & 5 \\ 0 & 0 & 6 & 0 & 6 & 6 \end{bmatrix}, r = 3$$

Rank r

Given : $A \in \mathbb{R}^{n \times n}$, $W \in \mathbb{R}^{n \times n}$, $r \in \mathbb{N}$, $k \in \mathbb{N}$, $\epsilon > 0$

$$W = \begin{bmatrix} 1 & 1 & 1 & 2 & 2 & 2 \\ 0 & 1 & 2 & 1 & 2 & 3 \\ 1 & 1 & 3 & 2 & 4 & 4 \\ 0 & 0 & 4 & 0 & 4 & 4 \\ 1 & 0 & 5 & 1 & 6 & 5 \\ 0 & 0 & 6 & 0 & 6 & 6 \end{bmatrix}, r = 3$$

Output :

Rank r

Given : $A \in \mathbb{R}^{n \times n}$, $W \in \mathbb{R}^{n \times n}$, $r \in \mathbb{N}$, $k \in \mathbb{N}$, $\epsilon > 0$

$$W = \begin{bmatrix} 1 & 1 & 1 & 2 & 2 & 2 \\ 0 & 1 & 2 & 1 & 2 & 3 \\ 1 & 1 & 3 & 2 & 4 & 4 \\ 0 & 0 & 4 & 0 & 4 & 4 \\ 1 & 0 & 5 & 1 & 6 & 5 \\ 0 & 0 & 6 & 0 & 6 & 6 \end{bmatrix}, r = 3$$

Output : rank- k $\hat{A} \in \mathbb{R}^{n \times n}$ s.t.

Rank r

Given : $A \in \mathbb{R}^{n \times n}$, $W \in \mathbb{R}^{n \times n}$, $r \in \mathbb{N}$, $k \in \mathbb{N}$, $\epsilon > 0$

$$W = \begin{bmatrix} 1 & 1 & 1 & 2 & 2 & 2 \\ 0 & 1 & 2 & 1 & 2 & 3 \\ 1 & 1 & 3 & 2 & 4 & 4 \\ 0 & 0 & 4 & 0 & 4 & 4 \\ 1 & 0 & 5 & 1 & 6 & 5 \\ 0 & 0 & 6 & 0 & 6 & 6 \end{bmatrix}, r = 3$$

Output : rank- k $\hat{A} \in \mathbb{R}^{n \times n}$ s.t.

$$\|W \circ (A - \hat{A})\|_F^2 \leq (1 + \epsilon) \min_{\text{rank-}k A'} \|W \circ (A - A')\|_F^2$$

Rank r

Given : $A \in \mathbb{R}^{n \times n}$, $W \in \mathbb{R}^{n \times n}$, $r \in \mathbb{N}$, $k \in \mathbb{N}$, $\epsilon > 0$

$$W = \begin{bmatrix} 1 & 1 & 1 & 2 & 2 & 2 \\ 0 & 1 & 2 & 1 & 2 & 3 \\ 1 & 1 & 3 & 2 & 4 & 4 \\ 0 & 0 & 4 & 0 & 4 & 4 \\ 1 & 0 & 5 & 1 & 6 & 5 \\ 0 & 0 & 6 & 0 & 6 & 6 \end{bmatrix}, r = 3$$

Output : rank- k $\hat{A} \in \mathbb{R}^{n \times n}$ s.t.

$$\|W \circ (A - \hat{A})\|_F^2 \leq (1 + \epsilon) \min_{\text{rank-}k A'} \sum_{i,j} W_{i,j}^2 (A_{i,j} - A'_{i,j})^2$$

Rank r

Given : $A \in \mathbb{R}^{n \times n}$, $W \in \mathbb{R}^{n \times n}$, $r \in \mathbb{N}$, $k \in \mathbb{N}$, $\epsilon > 0$

$$W = \begin{bmatrix} 1 & 1 & 1 & 2 & 2 & 2 \\ 0 & 1 & 2 & 1 & 2 & 3 \\ 1 & 1 & 3 & 2 & 4 & 4 \\ 0 & 0 & 4 & 0 & 4 & 4 \\ 1 & 0 & 5 & 1 & 6 & 5 \\ 0 & 0 & 6 & 0 & 6 & 6 \end{bmatrix}, r = 3$$

Output : rank- k $\hat{A} \in \mathbb{R}^{n \times n}$ s.t.

$$\|W \circ (A - \hat{A})\|_F^2 \leq (1 + \epsilon) \text{OPT}$$

Rank r

Given : $A \in \mathbb{R}^{n \times n}$, $W \in \mathbb{R}^{n \times n}$, $r \in \mathbb{N}$, $k \in \mathbb{N}$, $\epsilon > 0$

$$W = \begin{bmatrix} 1 & 1 & 1 & 2 & 2 & 2 \\ 0 & 1 & 2 & 1 & 2 & 3 \\ 1 & 1 & 3 & 2 & 4 & 4 \\ 0 & 0 & 4 & 0 & 4 & 4 \\ 1 & 0 & 5 & 1 & 6 & 5 \\ 0 & 0 & 6 & 0 & 6 & 6 \end{bmatrix}, r = 3$$

Output : rank- k $\hat{A} \in \mathbb{R}^{n \times n}$ s.t.

$$\|W \circ (A - \hat{A})\|_F^2 \leq (1 + \epsilon) \text{OPT}$$

with prob. 9/10

Rank r

Given : $A \in \mathbb{R}^{n \times n}$, $W \in \mathbb{R}^{n \times n}$, $r \in \mathbb{N}$, $k \in \mathbb{N}$, $\epsilon > 0$

$$W = \begin{bmatrix} 1 & 1 & 1 & 2 & 2 & 2 \\ 0 & 1 & 2 & 1 & 2 & 3 \\ 1 & 1 & 3 & 2 & 4 & 4 \\ 0 & 0 & 4 & 0 & 4 & 4 \\ 1 & 0 & 5 & 1 & 6 & 5 \\ 0 & 0 & 6 & 0 & 6 & 6 \end{bmatrix}, r = 3$$

Output : rank- k $\hat{A} \in \mathbb{R}^{n \times n}$ s.t.

$$\|W \circ (A - \hat{A})\|_F^2 \leq (1 + \epsilon) \text{OPT}$$

with prob. $9/10$

in $n^{O(k^2r/\epsilon)}$ time

Rank r

Given : $A \in \mathbb{R}^{n \times n}$, $W \in \mathbb{R}^{n \times n}$, $r \in \mathbb{N}$, $k \in \mathbb{N}$, $\epsilon > 0$

$$W = \begin{bmatrix} 1 & 1 & 1 & 2 & 2 & 2 \\ 0 & 1 & 2 & 1 & 2 & 3 \\ 1 & 1 & 3 & 2 & 4 & 4 \\ 0 & 0 & 4 & 0 & 4 & 4 \\ 1 & 0 & 5 & 1 & 6 & 5 \\ 0 & 0 & 6 & 0 & 6 & 6 \end{bmatrix}, r = 3$$

Output : rank- k $\hat{A} \in \mathbb{R}^{n \times n}$ s.t.

$$\|W \circ (A - \hat{A})\|_F^2 \leq (1 + \epsilon) \text{OPT}$$

with prob. 9/10

in $n^{O(k^2r/\epsilon)}$ time

previously only $r=1$ was known to be in polynomial time

Random-4SAT Hypothesis

Random-4SAT Hypothesis

[Feige'02, Goerdt-Lanka'04]

Random-4SAT Hypothesis

[Feige'02, Goerdt-Lanka'04]

Given : random 4-SAT formula \mathcal{S} on n variables

Random-4SAT Hypothesis

[Feige'02, Goerdt-Lanka'04]

Given : random 4-SAT formula \mathcal{S} on n variables
each clause has 4 literals

Random-4SAT Hypothesis

[Feige'02, Goerdt-Lanka'04]

Given : random 4-SAT formula \mathcal{S} on n variables
each clause has 4 literals
each of $\Theta(n^4)$ clauses is picked ind. with prob. $\Theta(1/n^3)$
 $m = \Theta(n)$ is the number of clauses

$$S_1 \quad (x_4 \vee x_{32} \vee x_9 \vee x_{20})$$

Random-4SAT Hypothesis

[Feige'02, Goerdt-Lanka'04]

Given : random 4-SAT formula \mathcal{S} on n variables
each clause has 4 literals
each of $\Theta(n^4)$ clauses is picked ind. with prob. $\Theta(1/n^3)$
 $m = \Theta(n)$ is the number of clauses

Any algorithm that :

$$\begin{array}{ll} S_1 & (x_4 \vee x_{32} \vee x_9 \vee x_{20}) \\ S_2 & (x_2 \vee x_{13} \vee x_5 \vee x_{25}) \\ & \dots\dots \\ S_m & (x_{11} \vee x_{19} \vee x_{24} \vee x_6) \end{array} \quad \mathcal{S} = S_1 \wedge S_2 \wedge \dots \wedge S_m$$

Random-4SAT Hypothesis

[Feige'02, Goerdt-Lanka'04]

Given : random 4-SAT formula \mathcal{S} on n variables
each clause has 4 literals
each of $\Theta(n^4)$ clauses is picked ind. with prob. $\Theta(1/n^3)$
 $m = \Theta(n)$ is the number of clauses

Any algorithm that :

outputs 1 with prob. 1 when \mathcal{S} is satisfiable
outputs 0 with prob. $\geq 1/2$
probability is over the input instances

$$S_1 \quad (x_4 \vee x_{32} \vee x_9 \vee x_{20})$$

$$S_2 \quad (x_2 \vee x_{13} \vee x_5 \vee x_{25}) \\ \dots \dots$$

$$\mathcal{S} = S_1 \wedge S_2 \wedge \dots \wedge S_m$$

$$S_m \quad (x_{11} \vee x_{19} \vee x_{24} \vee x_6)$$

Random-4SAT Hypothesis

[Feige'02, Goerdt-Lanka'04]

Given : random 4-SAT formula \mathcal{S} on n variables
each clause has 4 literals
each of $\Theta(n^4)$ clauses is picked ind. with prob. $\Theta(1/n^3)$
 $m = \Theta(n)$ is the number of clauses

Any algorithm that :

outputs 1 with prob. 1 when \mathcal{S} is satisfiable
outputs 0 with prob. $\geq 1/2$
probability is over the input instances

Requires : $2^{\Omega(n)}$ time

$$S_1 \quad (x_4 \vee x_{32} \vee x_9 \vee x_{20})$$

$$S_2 \quad (x_2 \vee x_{13} \vee x_5 \vee x_{25}) \\ \dots \dots$$

$$\mathcal{S} = S_1 \wedge S_2 \wedge \dots \wedge S_m$$

$$S_m \quad (x_{11} \vee x_{19} \vee x_{24} \vee x_6)$$

Maximum Edge Biclique(MEB)

Maximum Edge Biclique(MEB)

Given : A bipartite graph $G = (U, V, E)$ with $|U| = |V| = n$

Maximum Edge Biclique(MEB)

Given : A bipartite graph $G = (U, V, E)$ with $|U| = |V| = n$

Output : A k_1 -by- k_2 complete bipartite subgraph of G

Maximum Edge Biclique(MEB)

Given : A bipartite graph $G = (U, V, E)$ with $|U| = |V| = n$

Output : A k_1 -by- k_2 complete bipartite subgraph of G
s.t. $k_1 \cdot k_2$ is maximized

Maximum Edge Biclique(MEB)

Given : A bipartite graph $G = (U, V, E)$ with $|U| = |V| = n$

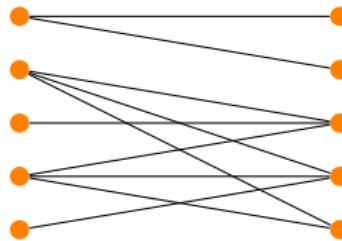
Output : A k_1 -by- k_2 complete bipartite subgraph of G
s.t. $k_1 \cdot k_2$ is maximized



Maximum Edge Biclique(MEB)

Given : A bipartite graph $G = (U, V, E)$ with $|U| = |V| = n$

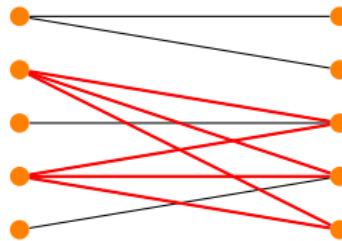
Output : A k_1 -by- k_2 complete bipartite subgraph of G
s.t. $k_1 \cdot k_2$ is maximized



Maximum Edge Biclique(MEB)

Given : A bipartite graph $G = (U, V, E)$ with $|U| = |V| = n$

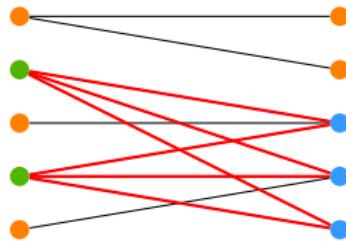
Output : A k_1 -by- k_2 complete bipartite subgraph of G
s.t. $k_1 \cdot k_2$ is maximized



Maximum Edge Biclique(MEB)

Given : A bipartite graph $G = (U, V, E)$ with $|U| = |V| = n$

Output : A k_1 -by- k_2 complete bipartite subgraph of G
s.t. $k_1 \cdot k_2$ is maximized



Maximum Edge Biclique(MEB)

[Goerdt-Lanka'04]

Maximum Edge Biclique(MEB)

[Goerdt-Lanka'04]

Assume : Random 4-SAT hardness hypothesis

Maximum Edge Biclique(MEB)

[Goerdt-Lanka'04]

Assume : Random 4-SAT hardness hypothesis

there exist two constants $\epsilon_1 > \epsilon_2 > 0$ such that

Maximum Edge Biclique(MEB)

[Goerdt-Lanka'04]

Assume : Random 4-SAT hardness hypothesis

there exist two constants $\epsilon_1 > \epsilon_2 > 0$ such that

Any algorithm that : distinguishes between

bipartite graphs $G = (U, V, E)$ with $|U| = |V| = n$ in two cases

Maximum Edge Biclique(MEB)

[Goerdt-Lanka'04]

Assume : Random 4-SAT hardness hypothesis

there exist two constants $\epsilon_1 > \epsilon_2 > 0$ such that

Any algorithm that : distinguishes between

bipartite graphs $G = (U, V, E)$ with $|U| = |V| = n$ in two cases

1. there is a clique of size $\geq (n/16)^2(1 + \epsilon_1)$

Maximum Edge Biclique(MEB)

[Goerdt-Lanka'04]

Assume : Random 4-SAT hardness hypothesis

there exist two constants $\epsilon_1 > \epsilon_2 > 0$ such that

Any algorithm that : distinguishes between

bipartite graphs $G = (U, V, E)$ with $|U| = |V| = n$ in two cases

1. there is a clique of size $\geq (n/16)^2(1 + \epsilon_1)$
2. all bipartite cliques are of size $\leq (n/16)^2(1 + \epsilon_2)$

Maximum Edge Biclique(MEB)

[Goerdt-Lanka'04]

Assume : Random 4-SAT hardness hypothesis

there exist two constants $\epsilon_1 > \epsilon_2 > 0$ such that

Any algorithm that : distinguishes between

bipartite graphs $G = (U, V, E)$ with $|U| = |V| = n$ in two cases

1. there is a clique of size $\geq (n/16)^2(1 + \epsilon_1)$
2. all bipartite cliques are of size $\leq (n/16)^2(1 + \epsilon_2)$

Requires :

Maximum Edge Biclique(MEB)

[Goerdt-Lanka'04]

Assume : Random 4-SAT hardness hypothesis

there exist two constants $\epsilon_1 > \epsilon_2 > 0$ such that

Any algorithm that : distinguishes between

bipartite graphs $G = (U, V, E)$ with $|U| = |V| = n$ in two cases

1. there is a clique of size $\geq (n/16)^2(1 + \epsilon_1)$
2. all bipartite cliques are of size $\leq (n/16)^2(1 + \epsilon_2)$

Requires : $2^{\Omega(n)}$ time

Reduction

Reduction

MEB

Reduction

MEB Given : U, V with $|U| = |V| = n$

Reduction

MEB Given : U, V with $|U| = |V| = n$
Output : a biclique s.t. $k_1 k_2$ is maximized

Reduction

MEB Given : U, V with $|U| = |V| = n$
Output : a biclique s.t. $k_1 k_2$ is maximized
 (complement) $|E| - k_1 k_2$ is minimized

Reduction

MEB Given : U, V with $|U| = |V| = n$
Output : a biclique s.t. $k_1 k_2$ is maximized
(complement) $|E| - k_1 k_2$ is minimized

WLRA

Reduction

MEB Given : U, V with $|U| = |V| = n$
Output : a biclique s.t. $k_1 k_2$ is maximized
(complement) $|E| - k_1 k_2$ is minimized

WLRA Given : $A \in \{0, 1\}^{n \times n}$, $W \in \mathbb{R}^{n \times n}$

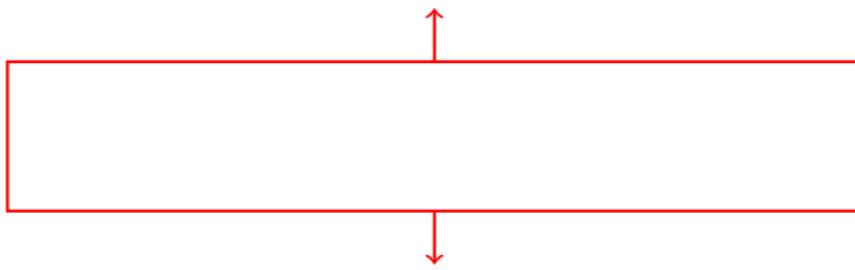
Reduction

MEB Given : U, V with $|U| = |V| = n$
Output : a biclique s.t. $k_1 k_2$ is maximized
(complement) $|E| - k_1 k_2$ is minimized

WLRA Given : $A \in \{0, 1\}^{n \times n}$, $W \in \mathbb{R}^{n \times n}$
Output : $u, v \in \mathbb{R}^n$ s.t. $\|W \circ (uv^\top - A)\|_F^2$ is minimized

Reduction

MEB Given : U, V with $|U| = |V| = n$
Output : a biclique s.t. $k_1 k_2$ is maximized
(complement) $|E| - k_1 k_2$ is minimized



WLRA Given : $A \in \{0, 1\}^{n \times n}$, $W \in \mathbb{R}^{n \times n}$
Output : $u, v \in \mathbb{R}^n$ s.t. $\|W \circ (uv^\top - A)\|_F^2$ is minimized

Reduction

MEB Given : U, V with $|U| = |V| = n$
Output : a biclique s.t. $k_1 k_2$ is maximized
(complement) $|E| - k_1 k_2$ is minimized



$$A_{i,j} = \begin{cases} 1 & \text{if edge } (U_i, V_j) \text{ exists} \\ 0 & \text{otherwise} \end{cases}$$



WLRA Given : $A \in \{0, 1\}^{n \times n}$, $W \in \mathbb{R}^{n \times n}$
Output : $u, v \in \mathbb{R}^n$ s.t. $\|W \circ (uv^\top - A)\|_F^2$ is minimized

Reduction

MEB Given : U, V with $|U| = |V| = n$
Output : a biclique s.t. $k_1 k_2$ is maximized
(complement) $|E| - k_1 k_2$ is minimized



$$A_{i,j} = \begin{cases} 1 \\ 0 \end{cases} \quad W_{i,j} = \begin{cases} 1 & \text{if edge } (U_i, V_j) \text{ exists} \\ n^6 & \text{otherwise} \end{cases}$$



WLRA Given : $A \in \{0, 1\}^{n \times n}$, $W \in \mathbb{R}^{n \times n}$
Output : $u, v \in \mathbb{R}^n$ s.t. $\|W \circ (uv^\top - A)\|_F^2$ is minimized

Main Results - Hardness

Main Results - Hardness

Weighted Low Rank Approximation Hardness

Main Results - Hardness

Weighted Low Rank Approximation Hardness

Given : $A \in \mathbb{R}^{n \times n}$, distinct r columns $W \in \mathbb{R}^{n \times n}$

Main Results - Hardness

Weighted Low Rank Approximation Hardness

Given : $A \in \mathbb{R}^{n \times n}$, distinct r columns $W \in \mathbb{R}^{n \times n}$
 $k \in \mathbb{N}, \epsilon > 0, W_{ij} \in \{0, 1, 2, \dots, \text{poly}(n)\}$

Main Results - Hardness

Weighted Low Rank Approximation Hardness

Given : $\textcolor{green}{A} \in \mathbb{R}^{n \times n}$, distinct r columns $\textcolor{blue}{W} \in \mathbb{R}^{n \times n}$
 $k \in \mathbb{N}$, $\epsilon > 0$, $\textcolor{blue}{W}_{ij} \in \{0, 1, 2, \dots, \text{poly}(n)\}$

Output : rank- k $\hat{\textcolor{red}{A}}$ s.t. $\|\textcolor{blue}{W} \circ (\textcolor{green}{A} - \hat{\textcolor{red}{A}})\|_F^2 \leq (1 + \epsilon) \text{OPT}$
with prob. $9/10$

Main Results - Hardness

Weighted Low Rank Approximation Hardness

Given : $A \in \mathbb{R}^{n \times n}$, distinct r columns $W \in \mathbb{R}^{n \times n}$
 $k \in \mathbb{N}$, $\epsilon > 0$, $W_{ij} \in \{0, 1, 2, \dots, \text{poly}(n)\}$

Output : rank- k \hat{A} s.t. $\|W \circ (A - \hat{A})\|_F^2 \leq (1 + \epsilon) \text{OPT}$
with prob. 9/10

Assume : Random-4SAT Hypothesis
[Feige'02, Goerdt-Lanka'04]

Main Results - Hardness

Weighted Low Rank Approximation Hardness

Given : $\textcolor{green}{A} \in \mathbb{R}^{n \times n}$, distinct r columns $\textcolor{blue}{W} \in \mathbb{R}^{n \times n}$
 $k \in \mathbb{N}$, $\epsilon > 0$, $\textcolor{blue}{W}_{ij} \in \{0, 1, 2, \dots, \text{poly}(n)\}$

Output : rank- k $\hat{\textcolor{red}{A}}$ s.t. $\|\textcolor{blue}{W} \circ (\textcolor{green}{A} - \hat{\textcolor{red}{A}})\|_F^2 \leq (1 + \epsilon) \text{OPT}$
with prob. 9/10

Assume : Random-4SAT Hypothesis
[Feige'02, Goerdt-Lanka'04]

$\exists \epsilon_0 > 0$, for any algorithm with $\epsilon < \epsilon_0$ and $k \geq 1$

Main Results - Hardness

Weighted Low Rank Approximation Hardness

Given : $\textcolor{green}{A} \in \mathbb{R}^{n \times n}$, distinct r columns $\textcolor{blue}{W} \in \mathbb{R}^{n \times n}$
 $k \in \mathbb{N}$, $\epsilon > 0$, $\textcolor{blue}{W}_{ij} \in \{0, 1, 2, \dots, \text{poly}(n)\}$

Output : rank- k $\hat{\textcolor{red}{A}}$ s.t. $\|\textcolor{blue}{W} \circ (\textcolor{green}{A} - \hat{\textcolor{red}{A}})\|_F^2 \leq (1 + \epsilon) \text{OPT}$
with prob. 9/10

Assume : Random-4SAT Hypothesis
[Feige'02, Goerdt-Lanka'04]

$\exists \epsilon_0 > 0$, for any algorithm with $\epsilon < \epsilon_0$ and $k \geq 1$

Requires : $2^{\Omega(r)}$ time

Algorithmic Techniques

- Tools

Algorithmic Techniques

- Tools

- ▶ Polynomial system verifier [Renegar'92, Basu-Pollack-Roy'96]

Algorithmic Techniques

• Tools

- ▶ Polynomial system verifier [Renegar'92, Basu-Pollack-Roy'96]
- ▶ Lower bound on the cost [Jeronimo-Perrucci-Tsigaridas'13]

Algorithmic Techniques

• Tools

- ▶ Polynomial system verifier [Renegar'92, Basu-Pollack-Roy'96]
- ▶ Lower bound on the cost [Jeronimo-Perrucci-Tsigaridas'13]
- ▶ Multiple regression sketch

Algorithmic Techniques

- Tools
 - ▶ Polynomial system verifier [Renegar'92, Basu-Pollack-Roy'96]
 - ▶ Lower bound on the cost [Jeronimo-Perrucci-Tsigaridas'13]
 - ▶ Multiple regression sketch
- Algorithm, assuming a rank r weight matrix \mathbf{W}

Algorithmic Techniques

- Tools
 - ▶ Polynomial system verifier [Renegar'92, Basu-Pollack-Roy'96]
 - ▶ Lower bound on the cost [Jeronimo-Perrucci-Tsigaridas'13]
 - ▶ Multiple regression sketch
- Algorithm, assuming a rank r weight matrix \mathbf{W}
 - ▶ Warmup, inefficient WLRA algorithm

Algorithmic Techniques

- Tools
 - ▶ Polynomial system verifier [Renegar'92, Basu-Pollack-Roy'96]
 - ▶ Lower bound on the cost [Jeronimo-Perrucci-Tsigaridas'13]
 - ▶ Multiple regression sketch
- Algorithm, assuming a rank r weight matrix \mathbf{W}
 - ▶ Warmup, inefficient WLRA algorithm
 - ▶ “Guessing a sketch”

Polynomial System Verifier (Recall)

Polynomial System Verifier (Recall)

[Renegar'92, Basu-Pollack-Roy'96]

Polynomial System Verifier (Recall)

[Renegar'92, Basu-Pollack-Roy'96]

Given : a real polynomial system $P(x)$

Polynomial System Verifier (Recall)

[Renegar'92, Basu-Pollack-Roy'96]

Given : a real polynomial system $P(x)$

v : #variables, $x = (x_1, x_2, \dots, x_v)$

Polynomial System Verifier (Recall)

[Renegar'92, Basu-Pollack-Roy'96]

Given : a real polynomial system $P(x)$

v : #variables, $x = (x_1, x_2, \dots, x_v)$

m : #polynomial constraints $f_i(x) \geq 0, \forall i \in [m]$

Polynomial System Verifier (Recall)

[Renegar'92, Basu-Pollack-Roy'96]

Given : a real polynomial system $P(x)$

v : #variables, $x = (x_1, x_2, \dots, x_v)$

m : #polynomial constraints $f_i(x) \geq 0, \forall i \in [m]$

d : maximum degree of all polynomial constraints

Polynomial System Verifier (Recall)

[Renegar'92, Basu-Pollack-Roy'96]

Given : a real polynomial system $P(x)$

v : #variables, $x = (x_1, x_2, \dots, x_v)$

m : #polynomial constraints $f_i(x) \geq 0, \forall i \in [m]$

d : maximum degree of all polynomial constraints

H : the bitsizes of the coefficients of the polynomials

Polynomial System Verifier (Recall)

[Renegar'92, Basu-Pollack-Roy'96]

Given : a real polynomial system $P(x)$

v : #variables, $x = (x_1, x_2, \dots, x_v)$

m : #polynomial constraints $f_i(x) \geq 0, \forall i \in [m]$

d : maximum degree of all polynomial constraints

H : the bitsizes of the coefficients of the polynomials

It takes $(md)^{O(v)}$ poly(H) time to
decide if there exists a solution to polynomial system P

Lower Bound on the Cost

Lower Bound on the Cost

[Jeronimo-Perrucci-Tsigaridas'13]

Lower Bound on the Cost

[Jeronimo-Perrucci-Tsigaridas'13]

Given: a real polynomial system $P(x)$

Lower Bound on the Cost

[Jeronimo-Perrucci-Tsigaridas'13]

Given: a real polynomial system $P(x)$

v : #variables, $x = (x_1, x_2, \dots, x_v)$

Lower Bound on the Cost

[Jeronimo-Perrucci-Tsigaridas'13]

Given: a real polynomial system $P(x)$

v : #variables, $x = (x_1, x_2, \dots, x_v)$

m : #polynomials constraints $f_i(x) \geq 0, \forall i \in [m]$

Lower Bound on the Cost

[Jeronimo-Perrucci-Tsigaridas'13]

Given: a real polynomial system $P(x)$

v : #variables, $x = (x_1, x_2, \dots, x_v)$

m : #polynomials constraints $f_i(x) \geq 0, \forall i \in [m]$

d : maximum degree of all polynomials

Lower Bound on the Cost

[Jeronimo-Perrucci-Tsigaridas'13]

Given: a real polynomial system $P(x)$

v : #variables, $x = (x_1, x_2, \dots, x_v)$

m : #polynomials constraints $f_i(x) \geq 0, \forall i \in [m]$

d : maximum degree of all polynomials

H : the bitsizes of the coefficients of the polynomials

Lower Bound on the Cost

[Jeronimo-Perrucci-Tsigaridas'13]

Given: a real polynomial system $P(x)$

v : #variables, $x = (x_1, x_2, \dots, x_v)$

m : #polynomials constraints $f_i(x) \geq 0, \forall i \in [m]$

d : maximum degree of all polynomials

H : the bitsizes of the coefficients of the polynomials



Lower Bound on the Cost

[Jeronimo-Perrucci-Tsigaridas'13]

Given: a real polynomial system $P(x)$

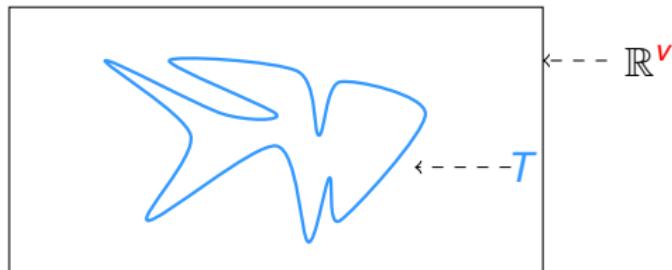
v : #variables, $x = (x_1, x_2, \dots, x_v)$

m : #polynomials constraints $f_i(x) \geq 0, \forall i \in [m]$

d : maximum degree of all polynomials

H : the bitsizes of the coefficients of the polynomials

T : $\{x \in \mathbb{R}^v \mid f_1(x) \geq 0, \dots, f_m(x) \geq 0\}$



Lower Bound on the Cost

[Jeronimo-Perrucci-Tsigaridas'13]

Given: a real polynomial system $P(x)$

v : #variables, $x = (x_1, x_2, \dots, x_v)$

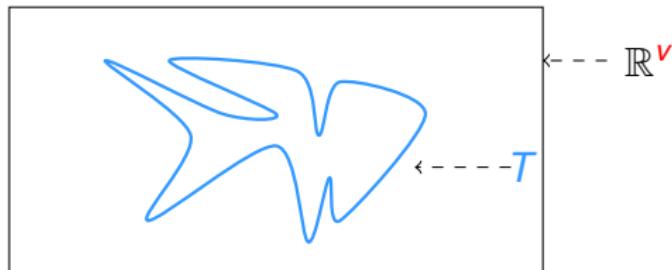
m : #polynomials constraints $f_i(x) \geq 0, \forall i \in [m]$

d : maximum degree of all polynomials

H : the bitsizes of the coefficients of the polynomials

T : $\{x \in \mathbb{R}^v \mid f_1(x) \geq 0, \dots, f_m(x) \geq 0\}$

minimum value that nonnegative g takes over $T \cap \text{Ball}$ is



Lower Bound on the Cost

[Jeronimo-Perrucci-Tsigaridas'13]

Given: a real polynomial system $P(x)$

v : #variables, $x = (x_1, x_2, \dots, x_v)$

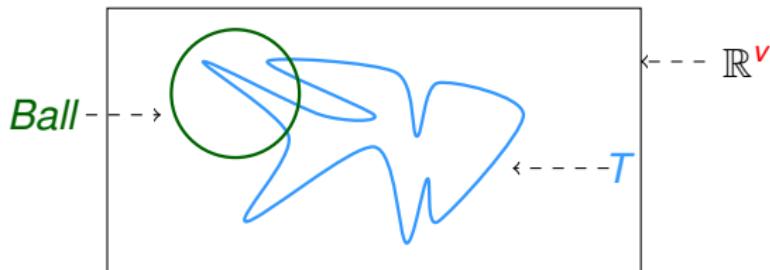
m : #polynomials constraints $f_i(x) \geq 0, \forall i \in [m]$

d : maximum degree of all polynomials

H : the bitsizes of the coefficients of the polynomials

T : $\{x \in \mathbb{R}^v \mid f_1(x) \geq 0, \dots, f_m(x) \geq 0\}$

minimum value that nonnegative g takes over $T \cap Ball$ is



Lower Bound on the Cost

[Jeronimo-Perrucci-Tsigaridas'13]

Given: a real polynomial system $P(x)$

v : #variables, $x = (x_1, x_2, \dots, x_v)$

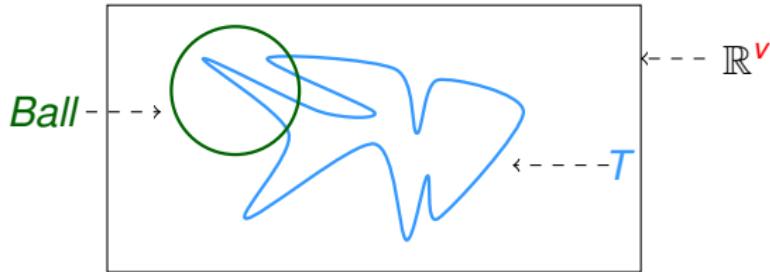
m : #polynomials constraints $f_i(x) \geq 0, \forall i \in [m]$

d : maximum degree of all polynomials

H : the bitsizes of the coefficients of the polynomials

T : $\{x \in \mathbb{R}^v \mid f_1(x) \geq 0, \dots, f_m(x) \geq 0\}$

minimum value that nonnegative g takes over $T \cap Ball$ is
either 0 or $\geq (2^H + m)^{-d^v}$



Lower Bound on the Cost

[Jeronimo-Perrucci-Tsigaridas'13]

Given: a real polynomial system $P(x)$

v : #variables, $x = (x_1, x_2, \dots, x_v)$

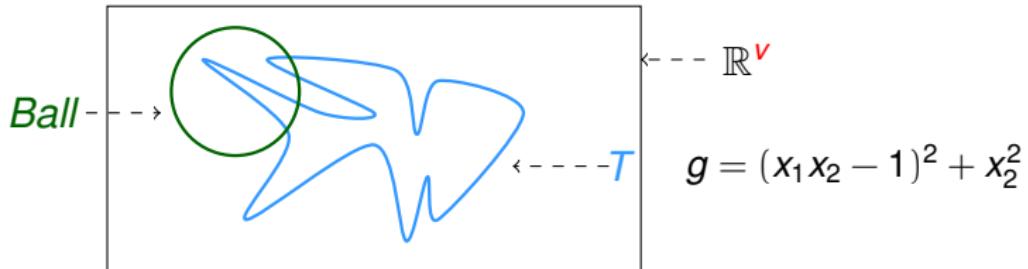
m : #polynomials constraints $f_i(x) \geq 0, \forall i \in [m]$

d : maximum degree of all polynomials

H : the bitsizes of the coefficients of the polynomials

T : $\{x \in \mathbb{R}^v \mid f_1(x) \geq 0, \dots, f_m(x) \geq 0\}$

minimum value that nonnegative g takes over $T \cap Ball$ is
either 0 or $\geq (2^H + m)^{-d^v}$



Multiple Regression Sketch

Multiple Regression Sketch

Given :

Multiple Regression Sketch

Given : $A^{(1)}, A^{(2)}, \dots, A^{(m)} \in \mathbb{R}^{n \times k}$

Multiple Regression Sketch

Given :

$$\mathbf{A}^{(1)}, \mathbf{A}^{(2)}, \dots, \mathbf{A}^{(m)} \in \mathbb{R}^{n \times k}$$

$$\mathbf{b}^{(1)}, \mathbf{b}^{(2)}, \dots, \mathbf{b}^{(m)} \in \mathbb{R}^{n \times 1}$$

Multiple Regression Sketch

Given : $\textcolor{green}{A}^{(1)}, \textcolor{green}{A}^{(2)}, \dots, \textcolor{green}{A}^{(m)} \in \mathbb{R}^{n \times k}$

$\textcolor{blue}{b}^{(1)}, \textcolor{blue}{b}^{(2)}, \dots, \textcolor{blue}{b}^{(m)} \in \mathbb{R}^{n \times 1}$

Let $\textcolor{red}{x}^{(j)} = \arg \min_{x \in \mathbb{R}^{k \times 1}} \|\textcolor{green}{A}^{(j)} x - \textcolor{blue}{b}^{(j)}\|, \forall j \in [m]$

Multiple Regression Sketch

Given : $\mathbf{A}^{(1)}, \mathbf{A}^{(2)}, \dots, \mathbf{A}^{(m)} \in \mathbb{R}^{n \times k}$

$\mathbf{b}^{(1)}, \mathbf{b}^{(2)}, \dots, \mathbf{b}^{(m)} \in \mathbb{R}^{n \times 1}$

Let $\mathbf{x}^{(j)} = \arg \min_{\mathbf{x} \in \mathbb{R}^{k \times 1}} \|\mathbf{A}^{(j)} \mathbf{x} - \mathbf{b}^{(j)}\|, \forall j \in [m]$

Choose : \mathbf{S} to be a random Gaussian matrix

Multiple Regression Sketch

Given : $\textcolor{green}{A}^{(1)}, \textcolor{green}{A}^{(2)}, \dots, \textcolor{green}{A}^{(m)} \in \mathbb{R}^{n \times k}$

$\textcolor{blue}{b}^{(1)}, \textcolor{blue}{b}^{(2)}, \dots, \textcolor{blue}{b}^{(m)} \in \mathbb{R}^{n \times 1}$

Let $\textcolor{red}{x}^{(j)} = \arg \min_{x \in \mathbb{R}^{k \times 1}} \|\textcolor{green}{A}^{(j)}x - \textcolor{blue}{b}^{(j)}\|, \forall j \in [m]$

Choose : $\textcolor{orange}{S}$ to be a random Gaussian matrix

Denote $\textcolor{red}{y}^{(j)} = \arg \min_{y \in \mathbb{R}^{k \times 1}} \|\textcolor{orange}{S}\textcolor{green}{A}^{(j)}y - \textcolor{orange}{S}\textcolor{blue}{b}^{(j)}\|, \forall j \in [m]$

Multiple Regression Sketch

Given : $\mathbf{A}^{(1)}, \mathbf{A}^{(2)}, \dots, \mathbf{A}^{(m)} \in \mathbb{R}^{n \times k}$

$\mathbf{b}^{(1)}, \mathbf{b}^{(2)}, \dots, \mathbf{b}^{(m)} \in \mathbb{R}^{n \times 1}$

Let $\mathbf{x}^{(j)} = \arg \min_{\mathbf{x} \in \mathbb{R}^{k \times 1}} \|\mathbf{A}^{(j)} \mathbf{x} - \mathbf{b}^{(j)}\|, \forall j \in [m]$

Choose : \mathbf{S} to be a random Gaussian matrix

Denote $\mathbf{y}^{(j)} = \arg \min_{\mathbf{y} \in \mathbb{R}^{k \times 1}} \|\mathbf{S} \mathbf{A}^{(j)} \mathbf{y} - \mathbf{S} \mathbf{b}^{(j)}\|, \forall j \in [m]$

Gaurantee :

Multiple Regression Sketch

Given : $\mathbf{A}^{(1)}, \mathbf{A}^{(2)}, \dots, \mathbf{A}^{(m)} \in \mathbb{R}^{n \times k}$

$\mathbf{b}^{(1)}, \mathbf{b}^{(2)}, \dots, \mathbf{b}^{(m)} \in \mathbb{R}^{n \times 1}$

Let $\mathbf{x}^{(j)} = \arg \min_{\mathbf{x} \in \mathbb{R}^{k \times 1}} \|\mathbf{A}^{(j)} \mathbf{x} - \mathbf{b}^{(j)}\|, \forall j \in [m]$

Choose : \mathbf{S} to be a random Gaussian matrix

Denote $\mathbf{y}^{(j)} = \arg \min_{\mathbf{y} \in \mathbb{R}^{k \times 1}} \|\mathbf{S} \mathbf{A}^{(j)} \mathbf{y} - \mathbf{S} \mathbf{b}^{(j)}\|, \forall j \in [m]$

Gaurantee : For all $\epsilon \in (0, 1/2)$, one can set $t = O(k/\epsilon)$ such that

Multiple Regression Sketch

Given : $\textcolor{green}{A}^{(1)}, \textcolor{green}{A}^{(2)}, \dots, \textcolor{green}{A}^{(m)} \in \mathbb{R}^{n \times k}$

$\textcolor{blue}{b}^{(1)}, \textcolor{blue}{b}^{(2)}, \dots, \textcolor{blue}{b}^{(m)} \in \mathbb{R}^{n \times 1}$

Let $\textcolor{red}{x}^{(j)} = \arg \min_{x \in \mathbb{R}^{k \times 1}} \|\textcolor{green}{A}^{(j)}x - \textcolor{blue}{b}^{(j)}\|, \forall j \in [m]$

Choose : $\textcolor{orange}{S}$ to be a random Gaussian matrix

Denote $\textcolor{red}{y}^{(j)} = \arg \min_{y \in \mathbb{R}^{k \times 1}} \|\textcolor{orange}{S}\textcolor{green}{A}^{(j)}y - \textcolor{orange}{S}\textcolor{blue}{b}^{(j)}\|, \forall j \in [m]$

Gaurantee : For all $\epsilon \in (0, 1/2)$, one can set $t = O(k/\epsilon)$ such that

$$\sum_{j=1}^m \|\textcolor{green}{A}^{(j)}\textcolor{red}{y}^{(j)} - \textcolor{blue}{b}^{(j)}\|_2^2 \leq (1 + \epsilon) \sum_{j=1}^m \|\textcolor{green}{A}^{(j)}\textcolor{red}{x}^{(j)} - \textcolor{blue}{b}^{(j)}\|_2^2$$

Multiple Regression Sketch

Given : $\mathbf{A}^{(1)}, \mathbf{A}^{(2)}, \dots, \mathbf{A}^{(m)} \in \mathbb{R}^{n \times k}$

$\mathbf{b}^{(1)}, \mathbf{b}^{(2)}, \dots, \mathbf{b}^{(m)} \in \mathbb{R}^{n \times 1}$

Let $\mathbf{x}^{(j)} = \arg \min_{\mathbf{x} \in \mathbb{R}^{k \times 1}} \|\mathbf{A}^{(j)} \mathbf{x} - \mathbf{b}^{(j)}\|, \forall j \in [m]$

Choose : \mathbf{S} to be a random Gaussian matrix

Denote $\mathbf{y}^{(j)} = \arg \min_{\mathbf{y} \in \mathbb{R}^{k \times 1}} \|\mathbf{S} \mathbf{A}^{(j)} \mathbf{y} - \mathbf{S} \mathbf{b}^{(j)}\|, \forall j \in [m]$

Gaurantee : For all $\epsilon \in (0, 1/2)$, one can set $t = O(k/\epsilon)$ such that

$$\sum_{j=1}^m \|\mathbf{A}^{(j)} \mathbf{y}^{(j)} - \mathbf{b}^{(j)}\|_2^2 \leq (1 + \epsilon) \sum_{j=1}^m \|\mathbf{A}^{(j)} \mathbf{x}^{(j)} - \mathbf{b}^{(j)}\|_2^2$$

with prob. 9/10

Multiple Regression Sketch

Given : $\mathbf{A}^{(1)}, \mathbf{A}^{(2)}, \dots, \mathbf{A}^{(m)} \in \mathbb{R}^{n \times k}$

$\mathbf{b}^{(1)}, \mathbf{b}^{(2)}, \dots, \mathbf{b}^{(m)} \in \mathbb{R}^{n \times 1}$

Let $\mathbf{x}^{(j)} = \arg \min_{\mathbf{x} \in \mathbb{R}^{k \times 1}} \|\mathbf{A}^{(j)} \mathbf{x} - \mathbf{b}^{(j)}\|, \forall j \in [m]$

Choose : \mathbf{S} to be a random Gaussian matrix

Denote $\mathbf{y}^{(j)} = \arg \min_{\mathbf{y} \in \mathbb{R}^{k \times 1}} \|\mathbf{S} \mathbf{A}^{(j)} \mathbf{y} - \mathbf{S} \mathbf{b}^{(j)}\|, \forall j \in [m]$

Gaurantee : For all $\epsilon \in (0, 1/2)$, one can set $t = O(k/\epsilon)$ such that

$$\sum_{j=1}^m \|\mathbf{A}^{(j)} \mathbf{y}^{(j)} - \mathbf{b}^{(j)}\|_2^2 \leq (1 + \epsilon) \sum_{j=1}^m \|\mathbf{A}^{(j)} \mathbf{x}^{(j)} - \mathbf{b}^{(j)}\|_2^2$$

with prob. 9/10

Warmup, inefficient WLRA Algorithm

Warmup, inefficient WLRA Algorithm

Given : $A \in \mathbb{R}^{n \times n}$, $W \in \mathbb{R}^{n \times n}$, $r \in \mathbb{N}$, $k \in \mathbb{N}$, $\epsilon > 0$

Warmup, inefficient WLRA Algorithm

Given : $A \in \mathbb{R}^{n \times n}$, $W \in \mathbb{R}^{n \times n}$, $r \in \mathbb{N}$, $k \in \mathbb{N}$, $\epsilon > 0$
 $A_{ij} \in \{0, \pm 1, \pm 2, \dots, \pm \Delta\}$

Warmup, inefficient WLRA Algorithm

Given : $A \in \mathbb{R}^{n \times n}$, $W \in \mathbb{R}^{n \times n}$, $r \in \mathbb{N}$, $k \in \mathbb{N}$, $\epsilon > 0$
 $W_{ij} \in \{0, 1, 2, \dots, \Delta\}$

Warmup, inefficient WLRA Algorithm

Given : $A \in \mathbb{R}^{n \times n}$, $W \in \mathbb{R}^{n \times n}$, $r \in \mathbb{N}$, $k \in \mathbb{N}$, $\epsilon > 0$

Output : rank- k \hat{A} s.t. $\|W \circ (A - \hat{A})\|_F^2 \leq (1 + \epsilon) \text{OPT}$

Warmup, inefficient WLRA Algorithm

Given : $A \in \mathbb{R}^{n \times n}$, $W \in \mathbb{R}^{n \times n}$, $r \in \mathbb{N}$, $k \in \mathbb{N}$, $\epsilon > 0$

Output : rank- k \hat{A} s.t. $\|W \circ (A - \hat{A})\|_F^2 \leq (1 + \epsilon) \text{OPT}$

Algorithm :

Warmup, inefficient WLRA Algorithm

Given : $\textcolor{green}{A} \in \mathbb{R}^{n \times n}$, $\textcolor{blue}{W} \in \mathbb{R}^{n \times n}$, $r \in \mathbb{N}$, $k \in \mathbb{N}$, $\epsilon > 0$

Output : rank- k $\hat{\textcolor{red}{A}}$ s.t. $\|\textcolor{blue}{W} \circ (\textcolor{green}{A} - \hat{\textcolor{red}{A}})\|_F^2 \leq (1 + \epsilon) \text{OPT}$

Algorithm :

1. create $2nk$ variables for $\textcolor{red}{U}, \textcolor{red}{V}^\top \in \mathbb{R}^{n \times k}$

Warmup, inefficient WLRA Algorithm

Given : $A \in \mathbb{R}^{n \times n}$, $W \in \mathbb{R}^{n \times n}$, $r \in \mathbb{N}$, $k \in \mathbb{N}$, $\epsilon > 0$

Output : rank- k \hat{A} s.t. $\|W \circ (A - \hat{A})\|_F^2 \leq (1 + \epsilon) \text{OPT}$

Algorithm :

1. create $2nk$ variables for $U, V^\top \in \mathbb{R}^{n \times k}$
2. write polynomial $g(x_1, \dots, x_{2nk}) = \|W \circ (A - UV)\|_F^2$

Warmup, inefficient WLRA Algorithm

Given : $A \in \mathbb{R}^{n \times n}$, $W \in \mathbb{R}^{n \times n}$, $r \in \mathbb{N}$, $k \in \mathbb{N}$, $\epsilon > 0$

Output : rank- k \hat{A} s.t. $\|W \circ (A - \hat{A})\|_F^2 \leq (1 + \epsilon) \text{OPT}$

Algorithm :

1. create $2nk$ variables for $U, V^\top \in \mathbb{R}^{n \times k}$
2. write polynomial $g(x_1, \dots, x_{2nk}) = \|W \circ (A - UV)\|_F^2$
3. pick $C \in [L^-, L^+]$, run polynomial verifier $g(x) \leq C$

Warmup, inefficient WLRA Algorithm

Given : $A \in \mathbb{R}^{n \times n}$, $W \in \mathbb{R}^{n \times n}$, $r \in \mathbb{N}$, $k \in \mathbb{N}$, $\epsilon > 0$

Output : rank- k \hat{A} s.t. $\|W \circ (A - \hat{A})\|_F^2 \leq (1 + \epsilon) \text{OPT}$

Algorithm :

1. create $2nk$ variables for $U, V^\top \in \mathbb{R}^{n \times k}$
2. write polynomial $g(x_1, \dots, x_{2nk}) = \|W \circ (A - UV)\|_F^2$
3. pick $C \in [L^-, L^+]$, run polynomial verifier $g(x) \leq C$
4. optimize C by binary search over $[L^-, L^+]$

Warmup, inefficient WLRA Algorithm

Given : $A \in \mathbb{R}^{n \times n}$, $W \in \mathbb{R}^{n \times n}$, $r \in \mathbb{N}$, $k \in \mathbb{N}$, $\epsilon > 0$

Output : rank- k \hat{A} s.t. $\|W \circ (A - \hat{A})\|_F^2 \leq (1 + \epsilon) \text{OPT}$

Algorithm :

1. create $2nk$ variables for $U, V^\top \in \mathbb{R}^{n \times k}$
2. write polynomial $g(x_1, \dots, x_{2nk}) = \|W \circ (A - UV)\|_F^2$
3. pick $C \in [L^-, L^+]$, run polynomial verifier $g(x) \leq C$
4. optimize C by binary search over $[L^-, L^+]$

Time : $2^{\Omega(nk)}$

Warmup, inefficient WLRA Algorithm

Given : $A \in \mathbb{R}^{n \times n}$, $W \in \mathbb{R}^{n \times n}$, $r \in \mathbb{N}$, $k \in \mathbb{N}$, $\epsilon > 0$

Output : rank- k \hat{A} s.t. $\|W \circ (A - \hat{A})\|_F^2 \leq (1 + \epsilon) \text{OPT}$

Algorithm :

1. create $2nk$ variables for $U, V^\top \in \mathbb{R}^{n \times k}$
2. write polynomial $g(x_1, \dots, x_{2nk}) = \|W \circ (A - UV)\|_F^2$
3. pick $C \in [L^-, L^+]$, run polynomial verifier $g(x) \leq C$
4. optimize C by binary search over $[L^-, L^+]$

Time :

$$2^{\Omega(nk)}$$

How can we do better?

Warmup, inefficient WLRA Algorithm

Given : $A \in \mathbb{R}^{n \times n}$, $W \in \mathbb{R}^{n \times n}$, $r \in \mathbb{N}$, $k \in \mathbb{N}$, $\epsilon > 0$

Output : rank- k \hat{A} s.t. $\|W \circ (A - \hat{A})\|_F^2 \leq (1 + \epsilon) \text{OPT}$

Algorithm :

1. create $2nk$ variables for $U, V^\top \in \mathbb{R}^{n \times k}$

Time : $2^{\Omega(nk)}$

How can we do better?

Warmup, inefficient WLRA Algorithm

Given : $A \in \mathbb{R}^{n \times n}$, $W \in \mathbb{R}^{n \times n}$, $r \in \mathbb{N}$, $k \in \mathbb{N}$, $\epsilon > 0$

Output : rank- k \hat{A} s.t. $\|W \circ (A - \hat{A})\|_F^2 \leq (1 + \epsilon) \text{OPT}$

Algorithm :

1. create $2nk$ variables for $U, V^\top \in \mathbb{R}^{n \times k}$

Time : $2^{\Omega(nk)}$

How can we do better?

polynomial verifier runs in $(\# \text{constraints} \cdot \text{degree})^{O(\# \text{variables})}$

Warmup, inefficient WLRA Algorithm

Given : $A \in \mathbb{R}^{n \times n}$, $W \in \mathbb{R}^{n \times n}$, $r \in \mathbb{N}$, $k \in \mathbb{N}$, $\epsilon > 0$

Output : rank- k \hat{A} s.t. $\|W \circ (A - \hat{A})\|_F^2 \leq (1 + \epsilon) \text{OPT}$

Algorithm :

1. create $2nk$ variables for $U, V^\top \in \mathbb{R}^{n \times k}$

Time : $2^{\Omega(nk)}$

How can we do better?

polynomial verifier runs in $(\# \text{constraints} \cdot \text{degree})^{O(\# \text{variables})}$
lower bound on cost $(\# \text{constraints})^{-\text{degree}^{O(\# \text{variables})}}$

Warmup, inefficient WLRA Algorithm

Given : $A \in \mathbb{R}^{n \times n}$, $W \in \mathbb{R}^{n \times n}$, $r \in \mathbb{N}$, $k \in \mathbb{N}$, $\epsilon > 0$

Output : rank- k \hat{A} s.t. $\|W \circ (A - \hat{A})\|_F^2 \leq (1 + \epsilon) \text{OPT}$

Algorithm :

1. create $2nk$ variables for $U, V^\top \in \mathbb{R}^{n \times k}$

Time : $2^{\Omega(nk)}$

How can we do better?

polynomial verifier runs in $(\# \text{constraints} \cdot \text{degree})^{O(\# \text{variables})}$

lower bound on cost $(\# \text{constraints})^{-\text{degree}}^{O(\# \text{variables})}$

write a polynomial with few **#variables**, i.e. $\text{poly}(kr/\epsilon)$

Warmup, inefficient WLRA Algorithm

Given : $A \in \mathbb{R}^{n \times n}$, $W \in \mathbb{R}^{n \times n}$, $r \in \mathbb{N}$, $k \in \mathbb{N}$, $\epsilon > 0$

Output : rank- k \hat{A} s.t. $\|W \circ (A - \hat{A})\|_F^2 \leq (1 + \epsilon) \text{OPT}$

Algorithm :

1. create $2nk$ variables for $U, V^\top \in \mathbb{R}^{n \times k}$

Time : $2^{\Omega(nk)}$

How can we do better?

polynomial verifier runs in $(\# \text{constraints} \cdot \text{degree})^{O(\# \text{variables})}$

lower bound on cost $(\# \text{constraints})^{-\text{degree}}^{O(\# \text{variables})}$

write a polynomial with few **#variables**, i.e. $\text{poly}(kr/\epsilon)$

without blowing up **degree** and **#constraints** too much

Main Idea

To reduce the number of variables to $\text{poly}(kr/\epsilon)$:

1. Multiple regression sketch with $O(k/\epsilon)$ rows
2. Weight matrix W has rank at most r

Guess a Sketch

Guess a Sketch

Given : $A \in \mathbb{R}^{n \times n}$, $W \in \mathbb{R}^{n \times n}$, $r \in \mathbb{N}$, $k \in \mathbb{N}$, $\epsilon > 0$

Guess a Sketch

Given : $A \in \mathbb{R}^{n \times n}$, $W \in \mathbb{R}^{n \times n}$, $r \in \mathbb{N}$, $k \in \mathbb{N}$, $\epsilon > 0$
 $A_{ij} \in \{0, \pm 1, \pm 2, \dots, \pm \Delta\}$

Guess a Sketch

Given : $A \in \mathbb{R}^{n \times n}$, $W \in \mathbb{R}^{n \times n}$, $r \in \mathbb{N}$, $k \in \mathbb{N}$, $\epsilon > 0$
 $W_{ij} \in \{0, 1, 2, \dots, \Delta\}$

Guess a Sketch

Given : $A \in \mathbb{R}^{n \times n}$, $W \in \mathbb{R}^{n \times n}$, $r \in \mathbb{N}$, $k \in \mathbb{N}$, $\epsilon > 0$

Output : $U, V^\top \in \mathbb{R}^{n \times k}$ s.t. $\|W \circ (UV - A)\|_F^2 \leq (1 + \epsilon) \text{OPT}$

Guess a Sketch

Given : $A \in \mathbb{R}^{n \times n}$, $W \in \mathbb{R}^{n \times n}$, $r \in \mathbb{N}$, $k \in \mathbb{N}$, $\epsilon > 0$

Output : $U, V^\top \in \mathbb{R}^{n \times k}$ s.t. $\|W \circ (UV - A)\|_F^2 \leq (1 + \epsilon) \text{OPT}$

Algorithm : W_j be j th column of W

D_{W_j} be diagonal matrix with vector W_j

$$W = \begin{bmatrix} 1 & 1 & 1 & 2 & 2 & 2 \\ 0 & 1 & 2 & 1 & 2 & 3 \\ 1 & 1 & 3 & 2 & 4 & 4 \\ 0 & 0 & 4 & 0 & 4 & 4 \\ 1 & 0 & 5 & 1 & 6 & 5 \\ 0 & 0 & 6 & 0 & 6 & 6 \end{bmatrix}$$

Guess a Sketch

Given : $A \in \mathbb{R}^{n \times n}$, $W \in \mathbb{R}^{n \times n}$, $r \in \mathbb{N}$, $k \in \mathbb{N}$, $\epsilon > 0$

Output : $U, V^\top \in \mathbb{R}^{n \times k}$ s.t. $\|W \circ (UV - A)\|_F^2 \leq (1 + \epsilon) \text{OPT}$

Algorithm : W_j be j th column of W

D_{W_j} be diagonal matrix with vector W_j

$$W = \begin{bmatrix} 1 & 1 & 1 & 2 & 2 & 2 \\ 0 & 1 & 2 & 1 & 2 & 3 \\ 1 & 1 & 3 & 2 & 4 & 4 \\ 0 & 0 & 4 & 0 & 4 & 4 \\ 1 & 0 & 5 & 1 & 6 & 5 \\ 0 & 0 & 6 & 0 & 6 & 6 \end{bmatrix} \quad D_{W_1} = \begin{bmatrix} 1 & & & & & \\ & 0 & & & & \\ & & 1 & & & \\ & & & 0 & & \\ & & & & 1 & \\ & & & & & 0 \end{bmatrix}$$

Guess a Sketch

Given : $A \in \mathbb{R}^{n \times n}$, $W \in \mathbb{R}^{n \times n}$, $r \in \mathbb{N}$, $k \in \mathbb{N}$, $\epsilon > 0$

Output : $U, V^\top \in \mathbb{R}^{n \times k}$ s.t. $\|W \circ (UV - A)\|_F^2 \leq (1 + \epsilon) \text{OPT}$

Algorithm : W_j be j th column of W

D_{W_j} be diagonal matrix with vector W_j

$$W = \begin{bmatrix} 1 & 1 & 2 & 2 & 2 \\ 0 & 1 & 2 & 1 & 2 & 3 \\ 1 & 1 & 3 & 2 & 4 & 4 \\ 0 & 0 & 4 & 0 & 4 & 4 \\ 1 & 0 & 5 & 1 & 6 & 5 \\ 0 & 0 & 6 & 0 & 6 & 6 \end{bmatrix} \quad D_{W_2} = \begin{bmatrix} 1 & & & & & \\ & 1 & & & & \\ & & 1 & & & \\ & & & 0 & & \\ & & & & 0 & \\ & & & & & 0 \end{bmatrix}$$

Guess a Sketch

Given : $A \in \mathbb{R}^{n \times n}$, $W \in \mathbb{R}^{n \times n}$, $r \in \mathbb{N}$, $k \in \mathbb{N}$, $\epsilon > 0$

Output : $U, V^\top \in \mathbb{R}^{n \times k}$ s.t. $\|W \circ (UV - A)\|_F^2 \leq (1 + \epsilon) \text{OPT}$

Algorithm : W_j be j th column of W

D_{W_j} be diagonal matrix with vector W_j

$$W = \begin{bmatrix} 1 & 1 & 1 & 2 & 2 & 2 \\ 0 & 1 & 2 & 1 & 2 & 3 \\ 1 & 1 & 3 & 2 & 4 & 4 \\ 0 & 0 & 4 & 0 & 4 & 4 \\ 1 & 0 & 5 & 1 & 6 & 5 \\ 0 & 0 & 6 & 0 & 6 & 6 \end{bmatrix} \quad D_{W_3} = \begin{bmatrix} 1 & & & & & \\ & 2 & & & & \\ & & 3 & & & \\ & & & 4 & & \\ & & & & 5 & \\ & & & & & 6 \end{bmatrix}$$

Guess a Sketch

Given : $A \in \mathbb{R}^{n \times n}$, $W \in \mathbb{R}^{n \times n}$, $r \in \mathbb{N}$, $k \in \mathbb{N}$, $\epsilon > 0$

Output : $U, V^\top \in \mathbb{R}^{n \times k}$ s.t. $\|W \circ (UV - A)\|_F^2 \leq (1 + \epsilon) \text{OPT}$

Algorithm : W_j be j th column of W

D_{W_j} be diagonal matrix with vector W_j

$$W = \begin{bmatrix} 1 & 1 & 1 & \boxed{2} & 2 & 2 \\ 0 & 1 & 2 & 1 & 2 & 3 \\ 1 & 1 & 3 & 2 & 4 & 4 \\ 0 & 0 & 4 & 0 & 4 & 4 \\ 1 & 0 & 5 & 1 & 6 & 5 \\ 0 & 0 & 6 & 0 & 6 & 6 \end{bmatrix} \quad D_{W_4} = \begin{bmatrix} 2 & & & & \\ & 1 & & & \\ & & 2 & & \\ & & & 0 & \\ & & & & 1 \\ & & & & & 0 \end{bmatrix}$$

Guess a Sketch

Given : $A \in \mathbb{R}^{n \times n}$, $W \in \mathbb{R}^{n \times n}$, $r \in \mathbb{N}$, $k \in \mathbb{N}$, $\epsilon > 0$

Output : $U, V^\top \in \mathbb{R}^{n \times k}$ s.t. $\|W \circ (UV - A)\|_F^2 \leq (1 + \epsilon) \text{OPT}$

Algorithm : W_j be j th column of W

D_{W_j} be diagonal matrix with vector W_j

$$W = \begin{bmatrix} 1 & 1 & 1 & 2 & 2 & 2 \\ 0 & 1 & 2 & 1 & 2 & 3 \\ 1 & 1 & 3 & 2 & 4 & 4 \\ 0 & 0 & 4 & 0 & 4 & 4 \\ 1 & 0 & 5 & 1 & 6 & 5 \\ 0 & 0 & 6 & 0 & 6 & 6 \end{bmatrix} \quad D_{W_5} = \begin{bmatrix} 2 & & & & & \\ & 2 & & & & \\ & & 4 & & & \\ & & & 4 & & \\ & & & & 6 & \\ & & & & & 6 \end{bmatrix}$$

Guess a Sketch

Given : $A \in \mathbb{R}^{n \times n}$, $W \in \mathbb{R}^{n \times n}$, $r \in \mathbb{N}$, $k \in \mathbb{N}$, $\epsilon > 0$

Output : $U, V^\top \in \mathbb{R}^{n \times k}$ s.t. $\|W \circ (UV - A)\|_F^2 \leq (1 + \epsilon) \text{OPT}$

Algorithm : W_j be j th column of W

D_{W_j} be diagonal matrix with vector W_j

$$W = \begin{bmatrix} 1 & 1 & 1 & 2 & 2 \\ 0 & 1 & 2 & 1 & 2 \\ 1 & 1 & 3 & 2 & 4 \\ 0 & 0 & 4 & 0 & 4 \\ 1 & 0 & 5 & 1 & 6 \\ 0 & 0 & 6 & 0 & 6 \end{bmatrix} \quad D_{W_6} = \begin{bmatrix} 2 & & & & \\ & 3 & & & \\ & & 4 & & \\ & & & 4 & \\ & & & & 5 \\ & & & & & 6 \end{bmatrix}$$

Guess a Sketch

Given : $A \in \mathbb{R}^{n \times n}$, $W \in \mathbb{R}^{n \times n}$, $r \in \mathbb{N}$, $k \in \mathbb{N}$, $\epsilon > 0$

Output : $U, V^\top \in \mathbb{R}^{n \times k}$ s.t. $\underbrace{\|W \circ (UV - A)\|_F^2}_{\leq (1 + \epsilon) \text{OPT}}$

Algorithm :

$$\left\| \begin{array}{c} W \\ \circ (\begin{array}{c} U \\ V \end{array}) - \begin{array}{c} A \end{array} \end{array} \right\|_F^2$$

Guess a Sketch

Given : $A \in \mathbb{R}^{n \times n}$, $W \in \mathbb{R}^{n \times n}$, $r \in \mathbb{N}$, $k \in \mathbb{N}$, $\epsilon > 0$

Output : $U, V^\top \in \mathbb{R}^{n \times k}$ s.t. $\underbrace{\|W \circ (UV - A)\|_F^2}_{\leftarrow} \leq (1 + \epsilon) \text{OPT}$

Algorithm :

$$\left\| \left(W \circ (UV) - A \right) \right\|_F^2$$

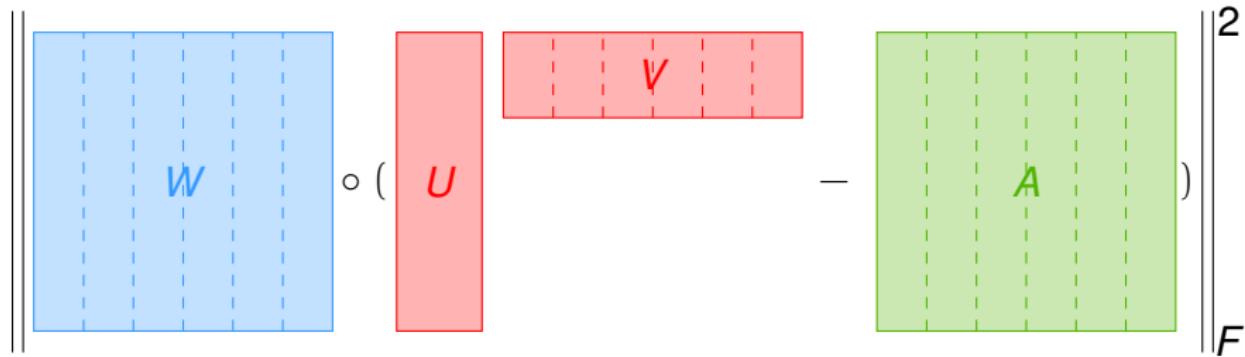
The diagram illustrates the computation of the Frobenius norm squared of the difference between the product of matrices W and UV , and the target matrix A . The matrices W , U , and V are represented as rectangles with vertical dashed lines, indicating their dimensions. The expression $W \circ (UV) - A$ is shown, where \circ denotes element-wise multiplication. The result of this expression is then squared and summed across all elements to get the Frobenius norm squared.

Guess a Sketch

Given : $A \in \mathbb{R}^{n \times n}$, $W \in \mathbb{R}^{n \times n}$, $r \in \mathbb{N}$, $k \in \mathbb{N}$, $\epsilon > 0$

Output : $U, V^\top \in \mathbb{R}^{n \times k}$ s.t. $\underbrace{\|W \circ (UV - A)\|_F^2}_{\leq (1 + \epsilon) \text{OPT}}$

Algorithm : $\sum_{j=1}^n \|D_{W_j} UV_j - D_{W_j} A_j\|_2^2 \leftarrow$

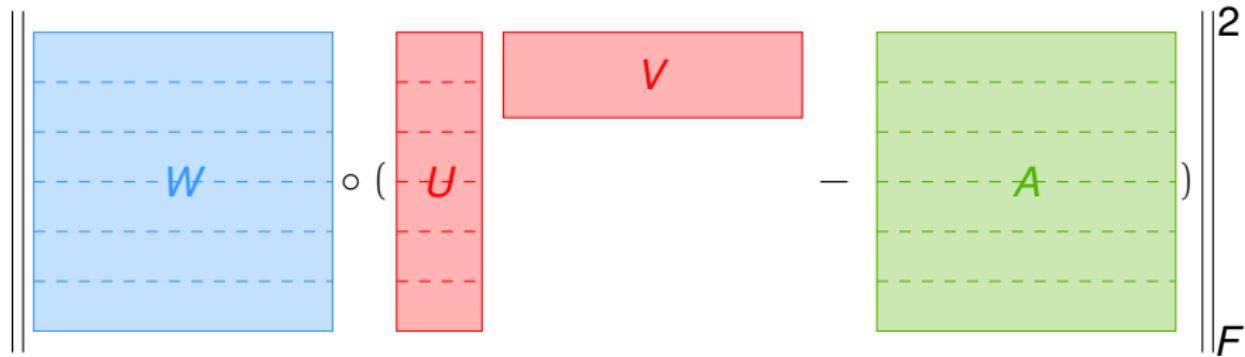


Guess a Sketch

Given : $A \in \mathbb{R}^{n \times n}$, $W \in \mathbb{R}^{n \times n}$, $r \in \mathbb{N}$, $k \in \mathbb{N}$, $\epsilon > 0$

Output : $U, V^\top \in \mathbb{R}^{n \times k}$ s.t. $\underbrace{\|W \circ (UV - A)\|_F^2}_{\leq (1 + \epsilon) \text{OPT}}$

Algorithm : $\sum_{j=1}^n \|D_{W_j} UV_j - D_{W_j} A_j\|_2^2 \leftarrow \rightarrow$

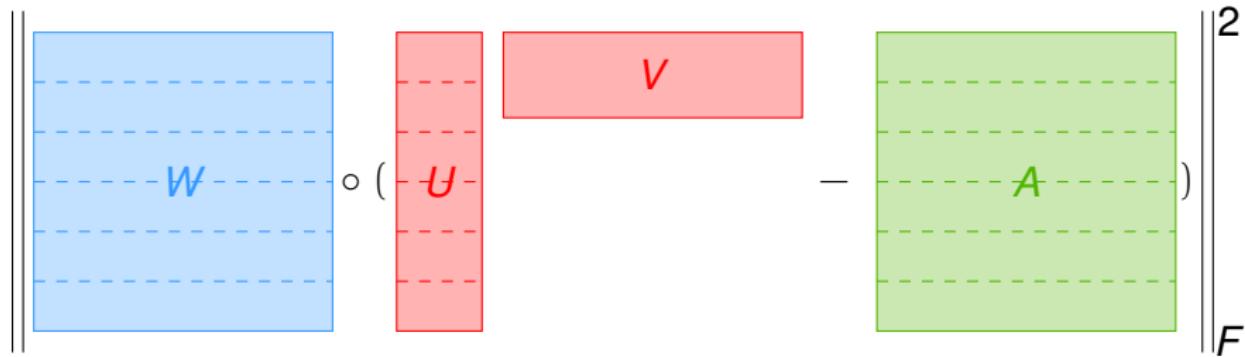


Guess a Sketch

Given : $A \in \mathbb{R}^{n \times n}$, $W \in \mathbb{R}^{n \times n}$, $r \in \mathbb{N}$, $k \in \mathbb{N}$, $\epsilon > 0$

Output : $U, V^\top \in \mathbb{R}^{n \times k}$ s.t. $\underbrace{\|W \circ (UV - A)\|_F^2}_{\leq (1 + \epsilon) \text{OPT}}$

Algorithm : $\sum_{j=1}^n \|D_{W_j} UV_j - D_{W_j} A_j\|_2^2 \leftarrow \sum_{i=1}^n \|U^i V D_{W_i} - A^i D_{W_i}\|_2^2$



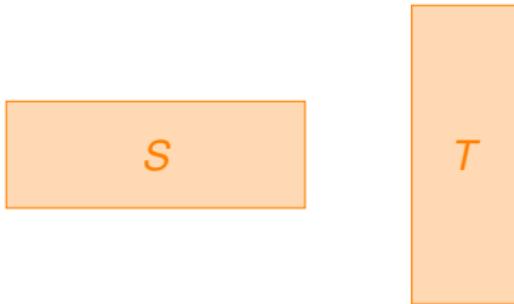
Guess a Sketch

Given : $A \in \mathbb{R}^{n \times n}$, $W \in \mathbb{R}^{n \times n}$, $r \in \mathbb{N}$, $k \in \mathbb{N}$, $\epsilon > 0$

Output : $U, V^\top \in \mathbb{R}^{n \times k}$ s.t. $\underbrace{\|W \circ (UV - A)\|_F^2}_{\text{OPT}}$ $\leq (1 + \epsilon) \text{OPT}$

Algorithm : $\sum_{j=1}^n \|D_{W_j} UV_j - D_{W_j} A_j\|_2^2 \leftarrow \sum_{i=1}^n \|U^i V D_{W_i} - A^i D_{W_i}\|_2^2$

Sketch by Gaussian $S, T^\top \in \mathbb{R}^{t \times n}$



Guess a Sketch

Given : $A \in \mathbb{R}^{n \times n}$, $W \in \mathbb{R}^{n \times n}$, $r \in \mathbb{N}$, $k \in \mathbb{N}$, $\epsilon > 0$

Output : $U, V^\top \in \mathbb{R}^{n \times k}$ s.t. $\underbrace{\|W \circ (UV - A)\|_F^2}_{\text{OPT}} \leq (1 + \epsilon) \text{OPT}$

Algorithm : $\sum_{j=1}^n \|D_{W_j} UV_j - D_{W_j} A_j\|_2^2 \leftarrow \sum_{i=1}^n \|U^i V D_{W_i} - A^i D_{W_i}\|_2^2$

↓ Sketch by Gaussian $S, T^\top \in \mathbb{R}^{t \times n}$

$$\sum_{j=1}^n \|SD_{W_j} UV_j - SD_{W_j} A_j\|_2^2$$

Guess a Sketch

Given : $A \in \mathbb{R}^{n \times n}$, $W \in \mathbb{R}^{n \times n}$, $r \in \mathbb{N}$, $k \in \mathbb{N}$, $\epsilon > 0$

Output : $U, V^\top \in \mathbb{R}^{n \times k}$ s.t. $\underbrace{\|W \circ (UV - A)\|_F^2}_{\text{OPT}} \leq (1 + \epsilon) \text{OPT}$

Algorithm : $\sum_{j=1}^n \|D_{W_j} UV_j - D_{W_j} A_j\|_2^2 \leftarrow \sum_{i=1}^n \|U^i V D_{W_i} - A^i D_{W_i}\|_2^2$

Sketch by Gaussian $S, T^\top \in \mathbb{R}^{t \times n}$

$$\sum_{j=1}^n \|SD_{W_j} UV_j - SD_{W_j} A_j\|_2^2 \quad \sum_{i=1}^n \|U^i V D_{W_i} T - A^i D_{W_i} T\|_2^2$$

Guess a Sketch

Given : $A \in \mathbb{R}^{n \times n}$, $W \in \mathbb{R}^{n \times n}$, $r \in \mathbb{N}$, $k \in \mathbb{N}$, $\epsilon > 0$

Output : $U, V^\top \in \mathbb{R}^{n \times k}$ s.t. $\underbrace{\|W \circ (UV - A)\|_F^2}_{\sum_{j=1}^n \|D_{W_j} UV_j - D_{W_j} A_j\|_2^2} \leq (1 + \epsilon) \text{OPT}$

Algorithm : $\sum_{j=1}^n \|D_{W_j} UV_j - D_{W_j} A_j\|_2^2 \leftarrow \sum_{i=1}^n \|U^i V D_{W_i} - A^i D_{W_i}\|_2^2$

↓ Sketch by Gaussian $S, T^\top \in \mathbb{R}^{t \times n}$ ↓
 $\sum_{j=1}^n \|SD_{W_j} UV_j - SD_{W_j} A_j\|_2^2$ $\sum_{i=1}^n \|U^i V D_{W_i} T - A^i D_{W_i} T\|_2^2$
Guess $SD_{W_j} U \in \mathbb{R}^{t \times k}$ and $VD_{W_i} T \in \mathbb{R}^{k \times t}$

Guess a Sketch

Given : $A \in \mathbb{R}^{n \times n}$, $W \in \mathbb{R}^{n \times n}$, $r \in \mathbb{N}$, $k \in \mathbb{N}$, $\epsilon > 0$

Output : $U, V^T \in \mathbb{R}^{n \times k}$ s.t. $\underbrace{\|W \circ (UV - A)\|_F^2}_{\sum_{j=1}^n \|D_{W_j} UV_j - D_{W_j} A_j\|_2^2} \leq (1 + \epsilon) \text{OPT}$

Algorithm : $\sum_{j=1}^n \|D_{W_j} UV_j - D_{W_j} A_j\|_2^2 \leftarrow \sum_{i=1}^n \|U^i V D_{W_i} - A^i D_{W_i}\|_2^2$

↓ Sketch by Gaussian $S, T^T \in \mathbb{R}^{t \times n}$ ↓
 $\sum_{j=1}^n \|SD_{W_j} UV_j - SD_{W_j} A_j\|_2^2$ $\sum_{i=1}^n \|U^i V D_{W_i} T - A^i D_{W_i} T\|_2^2$

Guess $SD_{W_1} U \in \mathbb{R}^{t \times k}$ and $VD_{W_i} T \in \mathbb{R}^{k \times t}$

$$SD_{W_1} U =$$

$$S \begin{bmatrix} 1 \\ 0 \\ 1 \\ 0 \\ 1 \\ 0 \end{bmatrix} = U$$

Guess a Sketch

Given : $A \in \mathbb{R}^{n \times n}$, $W \in \mathbb{R}^{n \times n}$, $r \in \mathbb{N}$, $k \in \mathbb{N}$, $\epsilon > 0$

Output : $U, V^T \in \mathbb{R}^{n \times k}$ s.t. $\underbrace{\|W \circ (UV - A)\|_F^2}_{\sum_{j=1}^n \|D_{W_j} UV_j - D_{W_j} A_j\|_2^2} \leq (1 + \epsilon) \text{OPT}$

Algorithm : $\sum_{j=1}^n \|D_{W_j} UV_j - D_{W_j} A_j\|_2^2 \leftarrow \sum_{i=1}^n \|U^i V D_{W_i} - A^i D_{W_i}\|_2^2$

↓ Sketch by Gaussian $S, T^T \in \mathbb{R}^{t \times n}$ ↓
 $\sum_{j=1}^n \|SD_{W_j} UV_j - SD_{W_j} A_j\|_2^2$ $\sum_{i=1}^n \|U^i V D_{W_i} T - A^i D_{W_i} T\|_2^2$

Guess $SD_{W_i} U \in \mathbb{R}^{t \times k}$ and $VD_{W_i} T \in \mathbb{R}^{k \times t}$

$$SD_{W_2} U =$$

$$S \begin{bmatrix} 1 \\ & 1 \\ & & 1 \\ & & & 0 \\ & & & & 0 \\ & & & & & 0 \end{bmatrix} U$$

Guess a Sketch

Given : $A \in \mathbb{R}^{n \times n}$, $W \in \mathbb{R}^{n \times n}$, $r \in \mathbb{N}$, $k \in \mathbb{N}$, $\epsilon > 0$

Output : $U, V^\top \in \mathbb{R}^{n \times k}$ s.t. $\underbrace{\|W \circ (UV - A)\|_F^2}_{\sum_{j=1}^n \|D_{W_j} UV_j - D_{W_j} A_j\|_2^2} \leq (1 + \epsilon) \text{OPT}$

Algorithm : $\sum_{j=1}^n \|D_{W_j} UV_j - D_{W_j} A_j\|_2^2 \leftarrow \sum_{i=1}^n \|U^i V D_{W_i} - A^i D_{W_i}\|_2^2$

↓ Sketch by Gaussian $S, T^\top \in \mathbb{R}^{t \times n}$ ↓
 $\sum_{j=1}^n \|SD_{W_j} UV_j - SD_{W_j} A_j\|_2^2$ $\sum_{i=1}^n \|U^i V D_{W_i} T - A^i D_{W_i} T\|_2^2$

Guess $SD_{W_i} U \in \mathbb{R}^{t \times k}$ and $VD_{W_i} T \in \mathbb{R}^{k \times t}$

$$SD_{W_3} U =$$

$$S \begin{bmatrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \end{bmatrix} U$$

Guess a Sketch

Given : $A \in \mathbb{R}^{n \times n}$, $W \in \mathbb{R}^{n \times n}$, $r \in \mathbb{N}$, $k \in \mathbb{N}$, $\epsilon > 0$

Output : $U, V^T \in \mathbb{R}^{n \times k}$ s.t. $\underbrace{\|W \circ (UV - A)\|_F^2}_{\sum_{j=1}^n \|D_{W_j} UV_j - D_{W_j} A_j\|_2^2} \leq (1 + \epsilon) \text{OPT}$

Algorithm : $\sum_{j=1}^n \|D_{W_j} UV_j - D_{W_j} A_j\|_2^2 \leftarrow \sum_{i=1}^n \|U^i V D_{W_i} - A^i D_{W_i}\|_2^2$

↓ Sketch by Gaussian $S, T^T \in \mathbb{R}^{t \times n}$ ↓
 $\sum_{j=1}^n \|SD_{W_j} UV_j - SD_{W_j} A_j\|_2^2$ $\sum_{i=1}^n \|U^i V D_{W_i} T - A^i D_{W_i} T\|_2^2$

Guess $SD_{W_i} U \in \mathbb{R}^{t \times k}$ and $VD_{W_i} T \in \mathbb{R}^{k \times t}$

$$SD_{W_4} U =$$

$$S \begin{bmatrix} 2 \\ 1 \\ 2 \\ 0 \\ 1 \\ 0 \end{bmatrix} = U$$

Guess a Sketch

Given : $A \in \mathbb{R}^{n \times n}$, $W \in \mathbb{R}^{n \times n}$, $r \in \mathbb{N}$, $k \in \mathbb{N}$, $\epsilon > 0$

Output : $U, V^T \in \mathbb{R}^{n \times k}$ s.t. $\underbrace{\|W \circ (UV - A)\|_F^2}_{\sum_{j=1}^n \|D_{W_j} UV_j - D_{W_j} A_j\|_2^2} \leq (1 + \epsilon) \text{OPT}$

Algorithm : $\sum_{j=1}^n \|D_{W_j} UV_j - D_{W_j} A_j\|_2^2 \leftarrow \sum_{i=1}^n \|U^i V D_{W_i} - A^i D_{W_i}\|_2^2$

↓ Sketch by Gaussian $S, T^T \in \mathbb{R}^{t \times n}$ ↓
 $\sum_{j=1}^n \|SD_{W_j} UV_j - SD_{W_j} A_j\|_2^2$ $\sum_{i=1}^n \|U^i V D_{W_i} T - A^i D_{W_i} T\|_2^2$

Guess $SD_{W_i} U \in \mathbb{R}^{t \times k}$ and $VD_{W_i} T \in \mathbb{R}^{k \times t}$

$$SD_{W_5} U =$$

$$S \begin{bmatrix} 2 \\ 2 \\ 4 \\ 4 \\ 6 \\ 6 \end{bmatrix} U$$

Guess a Sketch

Given : $A \in \mathbb{R}^{n \times n}$, $W \in \mathbb{R}^{n \times n}$, $r \in \mathbb{N}$, $k \in \mathbb{N}$, $\epsilon > 0$

Output : $U, V^T \in \mathbb{R}^{n \times k}$ s.t. $\underbrace{\|W \circ (UV - A)\|_F^2}_{\sum_{j=1}^n \|D_{W_j} UV_j - D_{W_j} A_j\|_2^2} \leq (1 + \epsilon) \text{OPT}$

Algorithm : $\sum_{j=1}^n \|D_{W_j} UV_j - D_{W_j} A_j\|_2^2 \leftarrow \sum_{i=1}^n \|U^i V D_{W_i} - A^i D_{W_i}\|_2^2$

↓ Sketch by Gaussian $S, T^T \in \mathbb{R}^{t \times n}$ ↓
 $\sum_{j=1}^n \|SD_{W_j} UV_j - SD_{W_j} A_j\|_2^2$ $\sum_{i=1}^n \|U^i V D_{W_i} T - A^i D_{W_i} T\|_2^2$

Guess $SD_{W_i} U \in \mathbb{R}^{t \times k}$ and $VD_{W_i} T \in \mathbb{R}^{k \times t}$

$$SD_{W_6} U =$$

$$S \begin{bmatrix} 2 \\ 3 \\ 4 \\ 4 \\ 5 \\ 6 \end{bmatrix} U$$

Guess a Sketch

Given : $A \in \mathbb{R}^{n \times n}$, $W \in \mathbb{R}^{n \times n}$, $r \in \mathbb{N}$, $k \in \mathbb{N}$, $\epsilon > 0$

Output : $U, V^\top \in \mathbb{R}^{n \times k}$ s.t. $\underbrace{\|W \circ (UV - A)\|_F^2}_{\sum_{j=1}^n \|D_{W_j} UV_j - D_{W_j} A_j\|_2^2} \leq (1 + \epsilon) \text{OPT}$

Algorithm : $\sum_{j=1}^n \|D_{W_j} UV_j - D_{W_j} A_j\|_2^2 \leftarrow \sum_{i=1}^n \|U^i V D_{W_i} - A^i D_{W_i}\|_2^2$

$$\begin{array}{ccc} \downarrow & \text{Sketch by Gaussian } S, T^\top \in \mathbb{R}^{t \times n} & \downarrow \\ \sum_{j=1}^n \|SD_{W_j} UV_j - SD_{W_j} A_j\|_2^2 & & \sum_{i=1}^n \|U^i V D_{W_i} T - A^i D_{W_i} T\|_2^2 \\ \text{Guess } SD_{W_j} U \in \mathbb{R}^{t \times k} \text{ and } V D_{W_i} T \in \mathbb{R}^{k \times t} & & \end{array}$$

Create $t \times k$ variables for each of $n SD_{W_j} U$ s?

Guess a Sketch

Given : $A \in \mathbb{R}^{n \times n}$, $W \in \mathbb{R}^{n \times n}$, $r \in \mathbb{N}$, $k \in \mathbb{N}$, $\epsilon > 0$

Output : $U, V^\top \in \mathbb{R}^{n \times k}$ s.t. $\underbrace{\|W \circ (UV - A)\|_F^2}_{\sum_{j=1}^n \|D_{W_j} UV_j - D_{W_j} A_j\|_2^2} \leq (1 + \epsilon) \text{OPT}$

Algorithm : $\sum_{j=1}^n \|D_{W_j} UV_j - D_{W_j} A_j\|_2^2 \leftarrow \sum_{i=1}^n \|U^i V D_{W_i} - A^i D_{W_i}\|_2^2$

$$\begin{array}{ccc} \downarrow & \text{Sketch by Gaussian } S, T^\top \in \mathbb{R}^{t \times n} & \downarrow \\ \sum_{j=1}^n \|SD_{W_j} UV_j - SD_{W_j} A_j\|_2^2 & & \sum_{i=1}^n \|U^i V D_{W_i} T - A^i D_{W_i} T\|_2^2 \\ \text{Guess } SD_{W_j} U \in \mathbb{R}^{t \times k} \text{ and } V D_{W_i} T \in \mathbb{R}^{k \times t} & & \end{array}$$

Create $t \times k$ variables for each of $n SD_{W_j} U$ s?

$n \times t \times k$

Guess a Sketch

Given : $A \in \mathbb{R}^{n \times n}$, $W \in \mathbb{R}^{n \times n}$, $r \in \mathbb{N}$, $k \in \mathbb{N}$, $\epsilon > 0$

Output : $U, V^\top \in \mathbb{R}^{n \times k}$ s.t. $\underbrace{\|W \circ (UV - A)\|_F^2}_{\sum_{j=1}^n \|D_{W_j} UV_j - D_{W_j} A_j\|_2^2} \leq (1 + \epsilon) \text{OPT}$

Algorithm : $\sum_{j=1}^n \|D_{W_j} UV_j - D_{W_j} A_j\|_2^2 \leftarrow \sum_{i=1}^n \|U^i V D_{W_i} - A^i D_{W_i}\|_2^2$

$$\begin{array}{ccc} \downarrow & \text{Sketch by Gaussian } S, T^\top \in \mathbb{R}^{t \times n} & \downarrow \\ \sum_{j=1}^n \|SD_{W_j} UV_j - SD_{W_j} A_j\|_2^2 & & \sum_{i=1}^n \|U^i V D_{W_i} T - A^i D_{W_i} T\|_2^2 \\ \text{Guess } SD_{W_j} U \in \mathbb{R}^{t \times k} \text{ and } V D_{W_i} T \in \mathbb{R}^{k \times t} & & \end{array}$$

Create $t \times k$ variables for each of $n SD_{W_j} U$ s?

? $\times t \times k$

Guess a Sketch

Given : $A \in \mathbb{R}^{n \times n}$, $W \in \mathbb{R}^{n \times n}$, $r \in \mathbb{N}$, $k \in \mathbb{N}$, $\epsilon > 0$

Output : $U, V^\top \in \mathbb{R}^{n \times k}$ s.t. $\underbrace{\|W \circ (UV - A)\|_F^2}_{\sum_{j=1}^n \|D_{W_j} UV_j - D_{W_j} A_j\|_2^2} \leq (1 + \epsilon) \text{OPT}$

Algorithm : $\sum_{j=1}^n \|D_{W_j} UV_j - D_{W_j} A_j\|_2^2 \leftarrow \sum_{i=1}^n \|U^i V D_{W_i} - A^i D_{W_i}\|_2^2$

$$\begin{array}{ccc} \downarrow & \text{Sketch by Gaussian } S, T^\top \in \mathbb{R}^{t \times n} & \downarrow \\ \sum_{j=1}^n \|SD_{W_j} UV_j - SD_{W_j} A_j\|_2^2 & & \sum_{i=1}^n \|U^i V D_{W_i} T - A^i D_{W_i} T\|_2^2 \\ \text{Guess } SD_{W_j} U \in \mathbb{R}^{t \times k} \text{ and } V D_{W_i} T \in \mathbb{R}^{k \times t} & & \end{array}$$

Create $t \times k$ variables for each of $n SD_{W_j} U$ s?

$r \times t \times k$

Guess a Sketch

Given : $A \in \mathbb{R}^{n \times n}$, $W \in \mathbb{R}^{n \times n}$, $r \in \mathbb{N}$, $k \in \mathbb{N}$, $\epsilon > 0$

Output : $U, V^\top \in \mathbb{R}^{n \times k}$ s.t. $\|W \circ (UV - A)\|_F^2 \leq (1 + \epsilon) \text{OPT}$

Algorithm : W_j be j th column of W

$$W = \begin{bmatrix} 1 & 1 & 1 & 2 & 2 & 2 \\ 0 & 1 & 2 & 1 & 2 & 3 \\ 1 & 1 & 3 & 2 & 4 & 4 \\ 0 & 0 & 4 & 0 & 4 & 4 \\ 1 & 0 & 5 & 1 & 6 & 5 \\ 0 & 0 & 6 & 0 & 6 & 6 \end{bmatrix}$$

Guess a Sketch

Given : $A \in \mathbb{R}^{n \times n}$, $W \in \mathbb{R}^{n \times n}$, $r \in \mathbb{N}$, $k \in \mathbb{N}$, $\epsilon > 0$

Output : $U, V^\top \in \mathbb{R}^{n \times k}$ s.t. $\|W \circ (UV - A)\|_F^2 \leq (1 + \epsilon) \text{OPT}$

Algorithm : W_j be j th column of W

column span of W

$$W = \left[\begin{array}{cccccc} 1 & 1 & 1 & 2 & 2 & 2 \\ 0 & 1 & 2 & 1 & 2 & 3 \\ 1 & 1 & 3 & 2 & 4 & 4 \\ 0 & 0 & 4 & 0 & 4 & 4 \\ 1 & 0 & 5 & 1 & 6 & 5 \\ 0 & 0 & 6 & 0 & 6 & 6 \end{array} \right] \quad \overbrace{\left[\begin{array}{ccc} 1 & 1 & 1 \\ 0 & 1 & 2 \\ 1 & 1 & 3 \\ 0 & 0 & 4 \\ 1 & 0 & 5 \\ 0 & 0 & 6 \end{array} \right]}$$

Guess a Sketch

Given : $A \in \mathbb{R}^{n \times n}$, $W \in \mathbb{R}^{n \times n}$, $r \in \mathbb{N}$, $k \in \mathbb{N}$, $\epsilon > 0$

Output : $U, V^\top \in \mathbb{R}^{n \times k}$ s.t. $\|W \circ (UV - A)\|_F^2 \leq (1 + \epsilon) \text{OPT}$

Algorithm : W_j be j th column of W

$$W_1 = W_1$$

$$W = \begin{bmatrix} 1 & 1 & 1 & 2 & 2 & 2 \\ 0 & 1 & 2 & 1 & 2 & 3 \\ 1 & 1 & 3 & 2 & 4 & 4 \\ 0 & 0 & 4 & 0 & 4 & 4 \\ 1 & 0 & 5 & 1 & 6 & 5 \\ 0 & 0 & 6 & 0 & 6 & 6 \end{bmatrix} \quad \begin{bmatrix} 1 \\ 0 \\ 1 \\ 0 \\ 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 1 \\ 0 \\ 1 \\ 0 \end{bmatrix}$$

Guess a Sketch

Given : $A \in \mathbb{R}^{n \times n}$, $W \in \mathbb{R}^{n \times n}$, $r \in \mathbb{N}$, $k \in \mathbb{N}$, $\epsilon > 0$

Output : $U, V^\top \in \mathbb{R}^{n \times k}$ s.t. $\|W \circ (UV - A)\|_F^2 \leq (1 + \epsilon) \text{OPT}$

Algorithm : W_j be j th column of W

$$W_2 = W_2$$

$$W = \begin{bmatrix} 1 & 1 & 1 & 2 & 2 & 2 \\ 0 & 1 & 2 & 1 & 2 & 3 \\ 1 & 1 & 3 & 2 & 4 & 4 \\ 0 & 0 & 4 & 0 & 4 & 4 \\ 1 & 0 & 5 & 1 & 6 & 5 \\ 0 & 0 & 6 & 0 & 6 & 6 \end{bmatrix} \quad \begin{bmatrix} 1 \\ 1 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

Guess a Sketch

Given : $A \in \mathbb{R}^{n \times n}$, $W \in \mathbb{R}^{n \times n}$, $r \in \mathbb{N}$, $k \in \mathbb{N}$, $\epsilon > 0$

Output : $U, V^\top \in \mathbb{R}^{n \times k}$ s.t. $\|W \circ (UV - A)\|_F^2 \leq (1 + \epsilon) \text{OPT}$

Algorithm : W_j be j th column of W

$$W = \begin{bmatrix} 1 & 1 & 1 & 2 & 2 & 2 \\ 0 & 1 & 2 & 1 & 2 & 3 \\ 1 & 1 & 3 & 2 & 4 & 4 \\ 0 & 0 & 4 & 0 & 4 & 4 \\ 1 & 0 & 5 & 1 & 6 & 5 \\ 0 & 0 & 6 & 0 & 6 & 6 \end{bmatrix} \quad W_3 = \begin{bmatrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \end{bmatrix}$$

Guess a Sketch

Given : $A \in \mathbb{R}^{n \times n}$, $W \in \mathbb{R}^{n \times n}$, $r \in \mathbb{N}$, $k \in \mathbb{N}$, $\epsilon > 0$

Output : $U, V^\top \in \mathbb{R}^{n \times k}$ s.t. $\|W \circ (UV - A)\|_F^2 \leq (1 + \epsilon) \text{OPT}$

Algorithm : W_j be j th column of W

$$W_4 = W_1 + W_2$$

$$W = \begin{bmatrix} 1 & 1 & 1 & \boxed{2} & 2 & 2 \\ 0 & 1 & 2 & 1 & 2 & 3 \\ 1 & 1 & 3 & 2 & 4 & 4 \\ 0 & 0 & 4 & 0 & 4 & 4 \\ 1 & 0 & 5 & 1 & 6 & 5 \\ 0 & 0 & 6 & 0 & 6 & 6 \end{bmatrix} \quad \begin{bmatrix} 2 \\ 1 \\ 2 \\ 0 \\ 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 1 \\ 0 \\ 1 \\ 0 \end{bmatrix} + \begin{bmatrix} 1 \\ 1 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

Guess a Sketch

Given : $A \in \mathbb{R}^{n \times n}$, $W \in \mathbb{R}^{n \times n}$, $r \in \mathbb{N}$, $k \in \mathbb{N}$, $\epsilon > 0$

Output : $U, V^\top \in \mathbb{R}^{n \times k}$ s.t. $\|W \circ (UV - A)\|_F^2 \leq (1 + \epsilon) \text{OPT}$

Algorithm : W_j be j th column of W

$$W_5 = W_1 + W_3$$

$$W = \begin{bmatrix} 1 & 1 & 1 & 2 & \boxed{2} & 2 \\ 0 & 1 & 2 & 1 & 2 & 3 \\ 1 & 1 & 3 & 2 & 4 & 4 \\ 0 & 0 & 4 & 0 & 4 & 4 \\ 1 & 0 & 5 & 1 & 6 & 5 \\ 0 & 0 & 6 & 0 & 6 & 6 \end{bmatrix} \quad \begin{bmatrix} 2 \\ 2 \\ 4 \\ 4 \\ 6 \\ 6 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 1 \\ 0 \\ 1 \\ 0 \end{bmatrix} + \begin{bmatrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \end{bmatrix}$$

Guess a Sketch

Given : $A \in \mathbb{R}^{n \times n}$, $W \in \mathbb{R}^{n \times n}$, $r \in \mathbb{N}$, $k \in \mathbb{N}$, $\epsilon > 0$

Output : $U, V^\top \in \mathbb{R}^{n \times k}$ s.t. $\|W \circ (UV - A)\|_F^2 \leq (1 + \epsilon) \text{OPT}$

Algorithm : W_j be j th column of W

$$W_6 = W_2 + W_3$$

$$W = \begin{bmatrix} 1 & 1 & 1 & 2 & 2 & 2 \\ 0 & 1 & 2 & 1 & 2 & 3 \\ 1 & 1 & 3 & 2 & 4 & 4 \\ 0 & 0 & 4 & 0 & 4 & 4 \\ 1 & 0 & 5 & 1 & 6 & 5 \\ 0 & 0 & 6 & 0 & 6 & 6 \end{bmatrix} \quad W_6 = \begin{bmatrix} 2 \\ 3 \\ 4 \\ 4 \\ 5 \\ 6 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \end{bmatrix}$$

Guess a Sketch

Given : $A \in \mathbb{R}^{n \times n}$, $W \in \mathbb{R}^{n \times n}$, $r \in \mathbb{N}$, $k \in \mathbb{N}$, $\epsilon > 0$

Output : $U, V^\top \in \mathbb{R}^{n \times k}$ s.t. $\underbrace{\|W \circ (UV - A)\|_F^2}_{\leq (1 + \epsilon) \text{OPT}}$

Algorithm :

$$\sum_{j=1}^n \|D_{W_j} UV_j - D_{W_j} A_j\|_2^2$$

Guess a Sketch

Given : $A \in \mathbb{R}^{n \times n}$, $W \in \mathbb{R}^{n \times n}$, $r \in \mathbb{N}$, $k \in \mathbb{N}$, $\epsilon > 0$

Output : $U, V^\top \in \mathbb{R}^{n \times k}$ s.t. $\underbrace{\|W \circ (UV - A)\|_F^2}_{\sum_{j=1}^n \|D_{W_j} UV_j - D_{W_j} A_j\|_2^2} \leq (1 + \epsilon) \text{OPT}$

Algorithm :

$$\text{sketch } \sum_{j=1}^n \|SD_{W_j} UV_j - SD_{W_j} A_j\|_2^2 \quad \begin{array}{l} \xrightarrow{\quad} \\ \xleftarrow{1 + \epsilon} \end{array}$$

Guess a Sketch

Given : $A \in \mathbb{R}^{n \times n}$, $W \in \mathbb{R}^{n \times n}$, $r \in \mathbb{N}$, $k \in \mathbb{N}$, $\epsilon > 0$

Output : $U, V^\top \in \mathbb{R}^{n \times k}$ s.t. $\underbrace{\|W \circ (UV - A)\|_F^2}_{\text{OPT}} \leq (1 + \epsilon) \text{OPT}$

Algorithm :

$$\begin{aligned} & \sum_{j=1}^n \|D_{W_j} UV_j - D_{W_j} A_j\|_2^2 \\ \text{sketch } & \sum_{j=1}^n \|SD_{W_j} UV_j - SD_{W_j} A_j\|_2^2 \quad \leftarrow 1 + \epsilon \\ \text{create variables for } & SD_{W_j} U, \forall j \in [r] \end{aligned}$$

Guess a Sketch

Given : $A \in \mathbb{R}^{n \times n}$, $W \in \mathbb{R}^{n \times n}$, $r \in \mathbb{N}$, $k \in \mathbb{N}$, $\epsilon > 0$

Output : $U, V^\top \in \mathbb{R}^{n \times k}$ s.t. $\underbrace{\|W \circ (UV - A)\|_F^2}_{\sum_{j=1}^n \|D_{W_j} UV_j - D_{W_j} A_j\|_2^2} \leq (1 + \epsilon) \text{OPT}$

Algorithm :

$\sum_{j=1}^n \|D_{W_j} UV_j - D_{W_j} A_j\|_2^2 \quad \boxed{1 + \epsilon}$

sketch $\sum_{j=1}^n \|SD_{W_j} UV_j - SD_{W_j} A_j\|_2^2 \quad \leftarrow$

create variables for $SD_{W_j} U$, $\forall j \in [r]$

create $t \times k$ variables for $SD_{W_1} U$

S

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

U

Guess a Sketch

Given : $A \in \mathbb{R}^{n \times n}$, $W \in \mathbb{R}^{n \times n}$, $r \in \mathbb{N}$, $k \in \mathbb{N}$, $\epsilon > 0$

Output : $U, V^\top \in \mathbb{R}^{n \times k}$ s.t. $\underbrace{\|W \circ (UV - A)\|_F^2}_{\sum_{j=1}^n \|D_{W_j} UV_j - D_{W_j} A_j\|_2^2} \leq (1 + \epsilon) \text{OPT}$

Algorithm :

sketch $\sum_{j=1}^n \|SD_{W_j} UV_j - SD_{W_j} A_j\|_2^2$ $\xrightarrow{1 + \epsilon}$
create variables for $SD_{W_j} U$, $\forall j \in [r]$
create $t \times k$ variables for $SD_{W_2} U$

S

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

U

Guess a Sketch

Given : $A \in \mathbb{R}^{n \times n}$, $W \in \mathbb{R}^{n \times n}$, $r \in \mathbb{N}$, $k \in \mathbb{N}$, $\epsilon > 0$

Output : $U, V^\top \in \mathbb{R}^{n \times k}$ s.t. $\underbrace{\|W \circ (UV - A)\|_F^2}_{\sum_{j=1}^n \|D_{W_j} UV_j - D_{W_j} A_j\|_2^2} \leq (1 + \epsilon) \text{OPT}$

Algorithm :

$\sum_{j=1}^n \|D_{W_j} UV_j - D_{W_j} A_j\|_2^2$ $1 + \epsilon$

sketch $\sum_{j=1}^n \|SD_{W_j} UV_j - SD_{W_j} A_j\|_2^2$

create variables for $SD_{W_j} U$, $\forall j \in [r]$

create $t \times k$ variables for $SD_{W_3} U$

S

$\begin{bmatrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \end{bmatrix}$

U

Guess a Sketch

Given : $A \in \mathbb{R}^{n \times n}$, $W \in \mathbb{R}^{n \times n}$, $r \in \mathbb{N}$, $k \in \mathbb{N}$, $\epsilon > 0$

Output : $U, V^\top \in \mathbb{R}^{n \times k}$ s.t. $\underbrace{\|W \circ (UV - A)\|_F^2}_n \leq (1 + \epsilon) \text{OPT}$

Algorithm :

$\sum_{j=1}^n \|D_{W_j} UV_j - D_{W_j} A_j\|_2^2$ $1 + \epsilon$

sketch $\sum_{j=1}^n \|SD_{W_j} UV_j - SD_{W_j} A_j\|_2^2$

create variables for $SD_{W_j} U$, $\forall j \in [r]$

write $SD_{W_4} U$ as $SD_{W_1} U + SD_{W_2} U$

S

$$\begin{bmatrix} 2 & & & \\ & 1 & & \\ & & 2 & \\ & & & 0 \\ & & & & 1 \\ & & & & & 0 \end{bmatrix}$$

U

Guess a Sketch

Given : $A \in \mathbb{R}^{n \times n}$, $W \in \mathbb{R}^{n \times n}$, $r \in \mathbb{N}$, $k \in \mathbb{N}$, $\epsilon > 0$

Output : $U, V^\top \in \mathbb{R}^{n \times k}$ s.t. $\underbrace{\|W \circ (UV - A)\|_F^2}_{\sum_{j=1}^n \|D_{W_j} UV_j - D_{W_j} A_j\|_2^2} \leq (1 + \epsilon) \text{OPT}$

Algorithm :

$\sum_{j=1}^n \|D_{W_j} UV_j - D_{W_j} A_j\|_2^2$ 1 + ϵ

sketch $\sum_{j=1}^n \|SD_{W_j} UV_j - SD_{W_j} A_j\|_2^2$ ←

create variables for $SD_{W_j} U$, $\forall j \in [r]$

write $SD_{W_4} U$ as $SD_{W_1} U + SD_{W_2} U$

$$S \begin{bmatrix} 1+1 & & & \\ & 0+1 & & \\ & & 1+1 & \\ & & & 0+0 \\ & & & & 1+0 \\ & & & & & 0+0 \end{bmatrix} U$$

Guess a Sketch

Given : $A \in \mathbb{R}^{n \times n}$, $W \in \mathbb{R}^{n \times n}$, $r \in \mathbb{N}$, $k \in \mathbb{N}$, $\epsilon > 0$

Output : $U, V^\top \in \mathbb{R}^{n \times k}$ s.t. $\underbrace{\|W \circ (UV - A)\|_F^2}_n \leq (1 + \epsilon) \text{OPT}$

Algorithm :

$$\begin{aligned} & \sum_{j=1}^n \|D_{W_j} UV_j - D_{W_j} A_j\|_2^2 \\ \text{sketch } & \sum_{j=1}^n \|SD_{W_j} UV_j - SD_{W_j} A_j\|_2^2 \quad \leftarrow 1 + \epsilon \\ \text{create variables for } & SD_{W_j} U, \forall j \in [r] \\ \text{write } & SD_{W_4} U \text{ as } SD_{W_1} U + SD_{W_2} U \end{aligned}$$

$$S \left(\begin{bmatrix} 1 & & & \\ & 0 & & \\ & & 1 & \\ & & & 0 \\ & & & & 1 \\ & & & & & 0 \end{bmatrix} + \begin{bmatrix} 1 & & & \\ & 1 & & \\ & & 1 & \\ & & & 0 \\ & & & & 0 \\ & & & & & 0 \end{bmatrix} \right) U$$

Guess a Sketch

Given : $A \in \mathbb{R}^{n \times n}$, $W \in \mathbb{R}^{n \times n}$, $r \in \mathbb{N}$, $k \in \mathbb{N}$, $\epsilon > 0$

Output : $U, V^\top \in \mathbb{R}^{n \times k}$ s.t. $\underbrace{\|W \circ (UV - A)\|_F^2}_n \leq (1 + \epsilon) \text{OPT}$

Algorithm :

$\sum_{j=1}^n \|D_{W_j} UV_j - D_{W_j} A_j\|_2^2$ $1 + \epsilon$

sketch $\sum_{j=1}^n \|SD_{W_j} UV_j - SD_{W_j} A_j\|_2^2$

create variables for $SD_{W_j} U$, $\forall j \in [r]$

write $SD_{W_5} U$ as $SD_{W_1} U + SD_{W_3} U$

S

$$\begin{bmatrix} 2 \\ 2 \\ 4 \\ 4 \\ 6 \\ 6 \end{bmatrix}$$

U

Guess a Sketch

Given : $A \in \mathbb{R}^{n \times n}$, $W \in \mathbb{R}^{n \times n}$, $r \in \mathbb{N}$, $k \in \mathbb{N}$, $\epsilon > 0$

Output : $U, V^\top \in \mathbb{R}^{n \times k}$ s.t. $\underbrace{\|W \circ (UV - A)\|_F^2}_{\sum_{j=1}^n \|D_{W_j} UV_j - D_{W_j} A_j\|_2^2} \leq (1 + \epsilon) \text{OPT}$

Algorithm :

$\sum_{j=1}^n \|D_{W_j} UV_j - D_{W_j} A_j\|_2^2$ 1 + ϵ

sketch $\sum_{j=1}^n \|SD_{W_j} UV_j - SD_{W_j} A_j\|_2^2$ ←

create variables for $SD_{W_j} U$, $\forall j \in [r]$

write $SD_{W_5} U$ as $SD_{W_1} U + SD_{W_3} U$

$$S \begin{bmatrix} 1+1 & & & & & \\ & 0+2 & & & & \\ & & 1+3 & & & \\ & & & 0+4 & & \\ & & & & 1+5 & \\ & & & & & 0+6 \end{bmatrix} U$$

Guess a Sketch

Given : $A \in \mathbb{R}^{n \times n}$, $W \in \mathbb{R}^{n \times n}$, $r \in \mathbb{N}$, $k \in \mathbb{N}$, $\epsilon > 0$

Output : $U, V^\top \in \mathbb{R}^{n \times k}$ s.t. $\underbrace{\|W \circ (UV - A)\|_F^2}_n \leq (1 + \epsilon) \text{OPT}$

Algorithm :

$$\begin{aligned} & \sum_{j=1}^n \|D_{W_j} UV_j - D_{W_j} A_j\|_2^2 \\ \text{sketch } & \sum_{j=1}^n \|SD_{W_j} UV_j - SD_{W_j} A_j\|_2^2 \quad \leftarrow 1 + \epsilon \\ \text{create variables for } & SD_{W_j} U, \forall j \in [r] \\ \text{write } & SD_{W_5} U \text{ as } SD_{W_1} U + SD_{W_3} U \end{aligned}$$

$$S \left(\begin{bmatrix} 1 & & & & & & \\ & 0 & & & & & \\ & & 1 & & & & \\ & & & 0 & & & \\ & & & & 1 & & \\ & & & & & 0 & \\ & & & & & & 1 \end{bmatrix} + \begin{bmatrix} 1 & & & & & & \\ & 2 & & & & & \\ & & 3 & & & & \\ & & & 4 & & & \\ & & & & 5 & & \\ & & & & & 6 & \end{bmatrix} \right) U$$

Guess a Sketch

Given : $A \in \mathbb{R}^{n \times n}$, $W \in \mathbb{R}^{n \times n}$, $r \in \mathbb{N}$, $k \in \mathbb{N}$, $\epsilon > 0$

Output : $U, V^\top \in \mathbb{R}^{n \times k}$ s.t. $\underbrace{\|W \circ (UV - A)\|_F^2}_n \leq (1 + \epsilon) \text{OPT}$

Algorithm :

$\sum_{j=1}^n \|D_{W_j} UV_j - D_{W_j} A_j\|_2^2$ $1 + \epsilon$

sketch $\sum_{j=1}^n \|SD_{W_j} UV_j - SD_{W_j} A_j\|_2^2$

create variables for $SD_{W_j} U$, $\forall j \in [r]$

write $SD_{W_6} U$ as $SD_{W_2} U + SD_{W_3} U$

S

$\begin{bmatrix} 2 \\ 3 \\ 4 \\ 4 \\ 5 \\ 6 \end{bmatrix}$

U

Guess a Sketch

Given : $A \in \mathbb{R}^{n \times n}$, $W \in \mathbb{R}^{n \times n}$, $r \in \mathbb{N}$, $k \in \mathbb{N}$, $\epsilon > 0$

Output : $U, V^\top \in \mathbb{R}^{n \times k}$ s.t. $\underbrace{\|W \circ (UV - A)\|_F^2}_{\sum_{j=1}^n \|D_{W_j} UV_j - D_{W_j} A_j\|_2^2} \leq (1 + \epsilon) \text{OPT}$

Algorithm :

$\sum_{j=1}^n \|D_{W_j} UV_j - D_{W_j} A_j\|_2^2 \quad \boxed{1 + \epsilon}$

sketch $\sum_{j=1}^n \|SD_{W_j} UV_j - SD_{W_j} A_j\|_2^2 \quad \leftarrow$

create variables for $SD_{W_j} U$, $\forall j \in [r]$

write $SD_{W_6} U$ as $SD_{W_2} U + SD_{W_3} U$

$$S \begin{bmatrix} 1+1 & & & & & \\ & 1+2 & & & & \\ & & 1+3 & & & \\ & & & 0+4 & & \\ & & & & 0+5 & \\ & & & & & 0+6 \end{bmatrix} U$$

Guess a Sketch

Given : $A \in \mathbb{R}^{n \times n}$, $W \in \mathbb{R}^{n \times n}$, $r \in \mathbb{N}$, $k \in \mathbb{N}$, $\epsilon > 0$

Output : $U, V^\top \in \mathbb{R}^{n \times k}$ s.t. $\underbrace{\|W \circ (UV - A)\|_F^2}_n \leq (1 + \epsilon) \text{OPT}$

Algorithm :

$$\begin{aligned} & \sum_{j=1}^n \|D_{W_j} UV_j - D_{W_j} A_j\|_2^2 \\ \text{sketch } & \sum_{j=1}^n \|SD_{W_j} UV_j - SD_{W_j} A_j\|_2^2 \quad \leftarrow 1 + \epsilon \\ \text{create variables for } & SD_{W_j} U, \forall j \in [r] \\ \text{write } & SD_{W_6} U \text{ as } SD_{W_2} U + SD_{W_3} U \end{aligned}$$

$$S \left(\begin{bmatrix} 1 & & & & & & \\ & 1 & & & & & \\ & & 1 & & & & \\ & & & 0 & & & \\ & & & & 0 & & \\ & & & & & 0 & \end{bmatrix} + \begin{bmatrix} 1 & & & & & & \\ & 2 & & & & & \\ & & 3 & & & & \\ & & & 4 & & & \\ & & & & 5 & & \\ & & & & & 6 & \end{bmatrix} \right) U$$

Wrapping Up

- Can use an explicit formula for the regression solution in terms of the variables created

Wrapping Up

- Can use an explicit formula for the regression solution in terms of the variables created
- The regression solution is a rational function, but can use tricks to clear the denominator

Wrapping Up

- Can use an explicit formula for the regression solution in terms of the variables created
- The regression solution is a rational function, but can use tricks to clear the denominator
- Multiple Regression Sketch + Bounded Rank Weight Matrix imply a small number of variables

Wrapping Up

- Can use an explicit formula for the regression solution in terms of the variables created
- The regression solution is a rational function, but can use tricks to clear the denominator
- Multiple Regression Sketch + Bounded Rank Weight Matrix imply a small number of variables
- Time : $n^{O(rk^2/\epsilon)}$

Wrapping Up

- Can use an explicit formula for the regression solution in terms of the variables created
- The regression solution is a rational function, but can use tricks to clear the denominator
- Multiple Regression Sketch + Bounded Rank Weight Matrix imply a small number of variables
- Time : $n^{O(rk^2/\epsilon)}$

$$(\# \text{constraints} \cdot \text{degree})^{O(\# \text{variables})} = (O(n) \cdot O(nk))^{O(krt)}$$

Open Problems

- For a rank- r weight matrix $\textcolor{blue}{W}$, the upper bound is $n^{O(k^2r/\epsilon)}$ but the lower bound is only $2^{\Omega(r)}$ - can we close this gap?

Open Problems

- For a rank- r weight matrix $\textcolor{blue}{W}$, the upper bound is $n^{O(k^2r/\epsilon)}$ but the lower bound is only $2^{\Omega(r)}$ - can we close this gap?
- Can we prove a hardness result with respect to the parameter k , e.g., a $2^{\Omega(k)}$ lower bound for WLRA problem?

ℓ_1 Low Rank Approximation

ℓ_1 -Low Rank Approximation

ℓ_1 -Low Rank Approximation

Given : $A \in \mathbb{R}^{n \times n}$, $k \in \mathbb{N}$, $\alpha \geq 1$

ℓ_1 -Low Rank Approximation

Given : $A \in \mathbb{R}^{n \times n}$, $k \in \mathbb{N}$, $\alpha \geq 1$

Output :

ℓ_1 -Low Rank Approximation

Given : $A \in \mathbb{R}^{n \times n}$, $k \in \mathbb{N}$, $\alpha \geq 1$

Output : rank- k $\hat{A} \in \mathbb{R}^{n \times n}$ s.t.

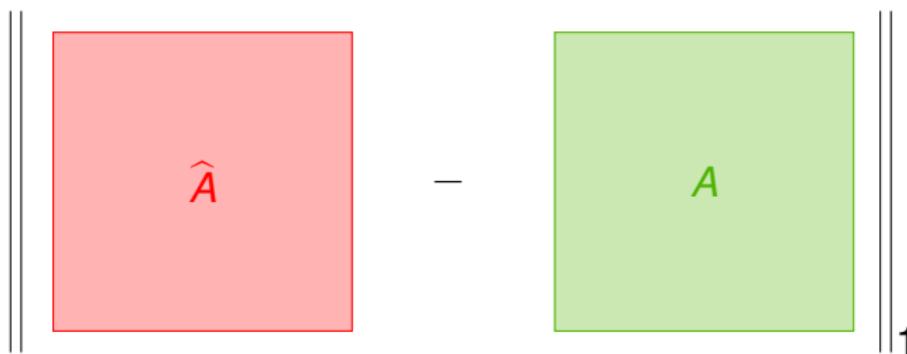
$$\left\| \hat{A} - A \right\|_1 \leq 1$$

ℓ_1 -Low Rank Approximation

Given : $A \in \mathbb{R}^{n \times n}$, $k \in \mathbb{N}$, $\alpha \geq 1$

Output : rank- k $\hat{A} \in \mathbb{R}^{n \times n}$ s.t.

$$\|\hat{A} - A\|_1 \leq \alpha \cdot \min_{\text{rank-}k A'} \|A' - A\|_1$$

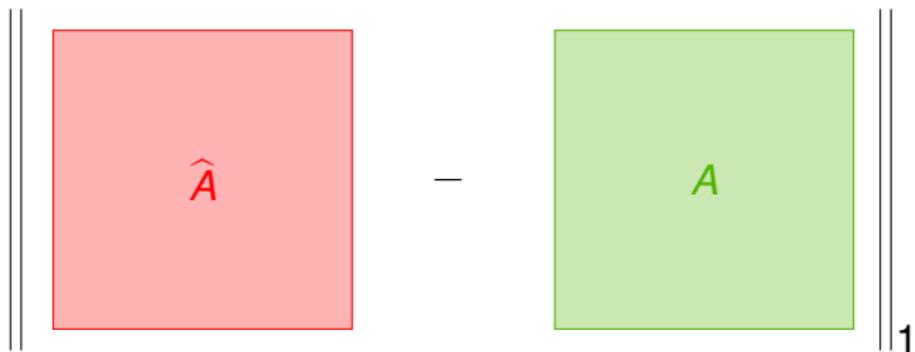


ℓ_1 -Low Rank Approximation

Given : $A \in \mathbb{R}^{n \times n}$, $k \in \mathbb{N}$, $\alpha \geq 1$

Output : rank- k $\hat{A} \in \mathbb{R}^{n \times n}$ s.t.

$$\|\hat{A} - A\|_1 \leq \alpha \cdot \underbrace{\min_{\text{rank-}k A'} \|\hat{A}' - A\|_1}_{\sum_{i,j} |A'_{i,j} - A_{i,j}|}$$



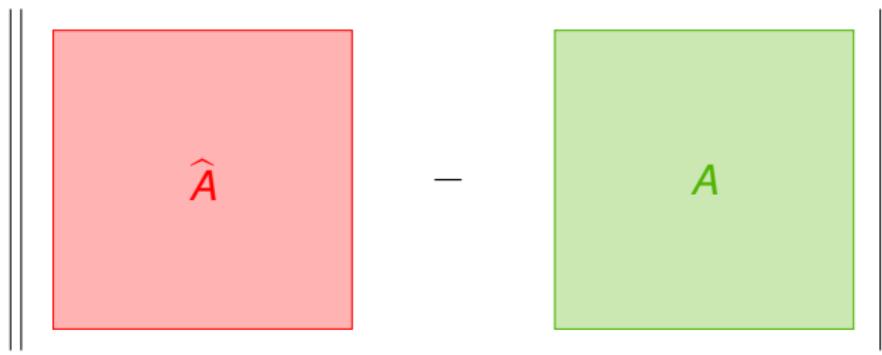
ℓ_1 -Low Rank Approximation

Given : $A \in \mathbb{R}^{n \times n}$, $k \in \mathbb{N}$, $\alpha \geq 1$

Output : rank- k $\hat{A} \in \mathbb{R}^{n \times n}$ s.t.

$$\|\hat{A} - A\|_1 \leq \alpha .$$

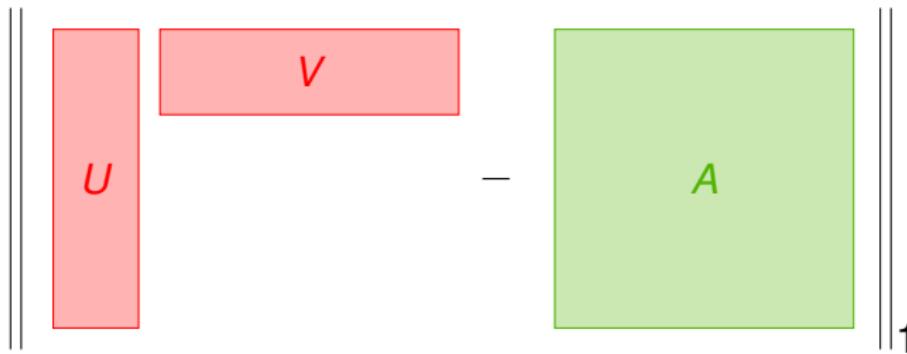
$$\underbrace{\min_{\text{rank-}k A'} \underbrace{\|A' - A\|_1}_{\sum_{i,j} |A'_{i,j} - A_{i,j}|}}_{\text{OPT}}$$



ℓ_1 -Low Rank Approximation

Given : $A \in \mathbb{R}^{n \times n}$, $k \in \mathbb{N}$, $\alpha \geq 1$

Output : $U \in \mathbb{R}^{n \times k}$, $V \in \mathbb{R}^{k \times n}$ s.t.
 $\|UV - A\|_1 \leq \alpha \cdot \text{OPT}$



Why is ℓ_1 -low Rank Approximation Interesting?

- The problem was shown to be NP-hard by Gillis-Vavasis'15 and a number of heuristics have been proposed

Why is ℓ_1 -low Rank Approximation Interesting?

- The problem was shown to be NP-hard by Gillis-Vavasis'15 and a number of heuristics have been proposed
- Asked in several places if there are any approximation algorithms

Thought Experiments

Thought Experiments

Given : A set of **points** in 2-dimensions

Thought Experiments

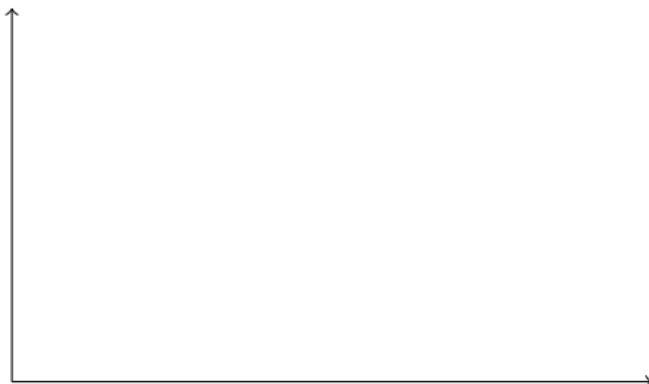
Given : A set of **points** in 2-dimensions

Goal : Find a line to fit those points

Thought Experiments

Given : A set of **points** in 2-dimensions

Goal : Find a line to fit those points



Thought Experiments

Given : A set of points in 2-dimensions

Goal : Find a line to fit those points

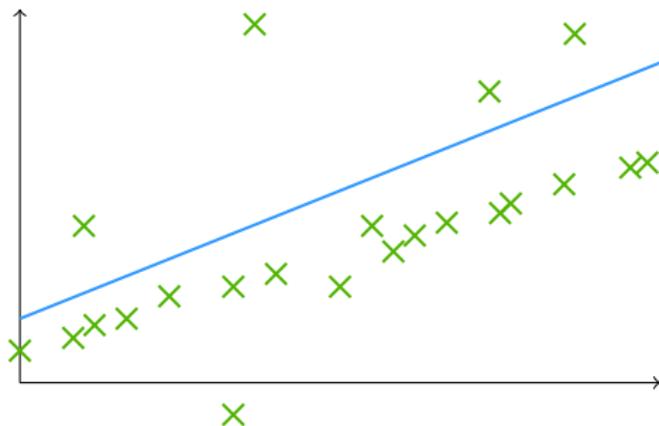


Thought Experiments

Given : A set of points in 2-dimensions

Goal : Find a line to fit those points

Output : ℓ_2 minimizer line,

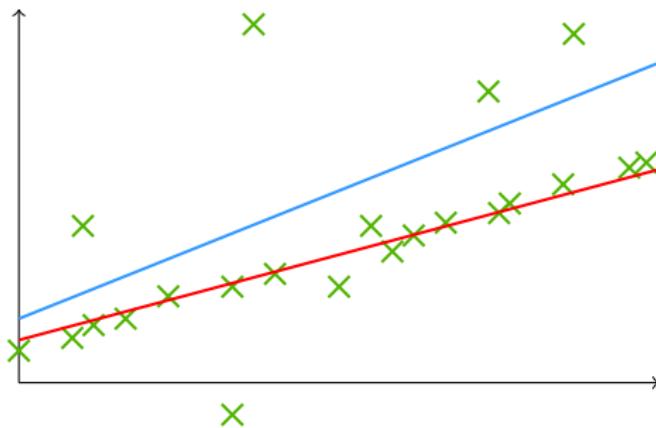


Thought Experiments

Given : A set of points in 2-dimensions

Goal : Find a line to fit those points

Output : ℓ_2 minimizer line, ℓ_1 minimizer line



Thought Experiments

Thought Experiments

Given : $k = 1$ and

$$A = \begin{bmatrix} 4 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 \end{bmatrix}$$

Thought Experiments

Given : $k = 1$ and

$$A = \begin{bmatrix} 4 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 \end{bmatrix}$$

Output : rank-1 $\|\cdot\|_F$ -solution $\hat{A} =$

$$\hat{A} = \begin{bmatrix} 4 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

Thought Experiments

Given : $k = 1$ and

$$A = \begin{bmatrix} 4 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 \end{bmatrix}$$

Output : rank-1 $\|\cdot\|_F$ -solution $\hat{A} =$

$$\begin{bmatrix} 4 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$



Thought Experiments

Given : $k = 1$ and

$$A = \begin{bmatrix} 4 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 \end{bmatrix}$$

Output : rank-1 $\|\cdot\|_F$ -solution $\hat{A} =$

$$\begin{bmatrix} 4 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$



Output : rank-1 $\|\cdot\|_1$ -solution $\hat{A} =$

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

Thought Experiments

Given : $k = 1$ and

$$A = \begin{bmatrix} 4 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 \end{bmatrix}$$

Output : rank-1 $\|\cdot\|_F$ -solution $\hat{A} =$

$$\begin{bmatrix} 4 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$



Output : rank-1 $\|\cdot\|_1$ -solution $\hat{A} =$

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



Heuristics and Nearly ℓ_1 Setting

- ℓ_1 -setting

Heuristics and Nearly ℓ_1 Setting

- ℓ_1 -setting
 - ▶ Heuristic algorithms Ke-Kanade'05, Ding-Zhou-He-Zha'06, Kwak'08, Brooks-Dulá-Boone'13

Heuristics and Nearly ℓ_1 Setting

- ℓ_1 -setting

- ▶ Heuristic algorithms Ke-Kanade'05, Ding-Zhou-He-Zha'06, Kwak'08, Brooks-Dulá-Boone'13
- ▶ Robust PCA Candès-Li-Ma-Wright'11, Wright-Ganesh-Rao-Peng-Ma'09, Netrapalli-Niranjan-Sanghavi-Anandkumar-Jain'14, Yi-Park-Chen-Caramanis'16

Heuristics and Nearly ℓ_1 Setting

- ℓ_1 -setting
 - ▶ Heuristic algorithms Ke-Kanade'05, Ding-Zhou-He-Zha'06, Kwak'08, Brooks-Dulá-Boone'13
 - ▶ Robust PCA Candès-Li-Ma-Wright'11, Wright-Ganesh-Rao-Peng-Ma'09, Netrapalli-Niranjan-Sanghavi-Anandkumar-Jain'14, Yi-Park-Chen-Caramanis'16
- None of them can achieve better than $\text{poly}(n)$ approximation ratio

Main Question, ℓ_1 Low Rank Approximation

Question

Given matrix $A \in \mathbb{R}^{n \times n}$, is there an algorithm that is able to output a rank- k matrix \hat{A} such that

$$\|\hat{A} - A\|_1 \leq \alpha \cdot \min_{\text{rank-}k A'} \|A' - A\|_1?$$

Or, are there any inapproximability results?

Main Results - Algorithms

- Upper bound (Algorithm) for an arbitrary $n \times n$ matrix A

Main Results - Algorithms

- Upper bound (Algorithm) for an arbitrary $n \times n$ matrix A
 - ▶ poly(k) log n -approximation, poly(n) time

Main Results - Algorithms

- Upper bound (Algorithm) for an arbitrary $n \times n$ matrix \mathbf{A}
 - ▶ poly(k) log n -approximation, poly(n) time
 - ▶ $\tilde{O}(k)$ -approximation, $n^{\tilde{O}(k)} 2^{\tilde{O}(k^2)}$ time.

Main Results - Algorithms

- Upper bound (Algorithm) for an arbitrary $n \times n$ matrix \mathbf{A}
 - ▶ poly(k) log n -approximation, poly(n) time
 - ▶ $\tilde{O}(k)$ -approximation, $n^{\tilde{O}(k)} 2^{\tilde{O}(k^2)}$ time.
 - ▶ Bicriteria algorithm

Main Results - Algorithms

- Upper bound (Algorithm) for an arbitrary $n \times n$ matrix \mathbf{A}
 - ▶ poly(k) log n -approximation, poly(n) time
 - ▶ $\tilde{O}(k)$ -approximation, $n^{\tilde{O}(k)} 2^{\tilde{O}(k^2)}$ time.
 - ▶ Bicriteria algorithm
 - ★ rank- $2k$ $O(1)$ -approximation, $n^{\tilde{O}(k^2)}$ running time

Hardness - Under Exponential Time Hypothesis

Hardness - Under Exponential Time Hypothesis

ℓ_1 -Low Rank Approximation Hardness

Hardness - Under Exponential Time Hypothesis

ℓ_1 -Low Rank Approximation Hardness

Given :

$$A \in \mathbb{R}^{n \times n}, k \in \mathbb{N}, \alpha > 1$$

Hardness - Under Exponential Time Hypothesis

ℓ_1 -Low Rank Approximation Hardness

Given : $A \in \mathbb{R}^{n \times n}$, $k \in \mathbb{N}$, $\alpha > 1$

Output : $U \in \mathbb{R}^{n \times k}$, $V \in \mathbb{R}^{k \times n}$ s.t. $\|UV - A\|_1 \leq \alpha \cdot \text{OPT}$
with prob. $9/10$

Hardness - Under Exponential Time Hypothesis

ℓ_1 -Low Rank Approximation Hardness

Given : $A \in \mathbb{R}^{n \times n}, k \in \mathbb{N}, \alpha > 1$

Output : $U \in \mathbb{R}^{n \times k}, V \in \mathbb{R}^{k \times n}$ s.t. $\|UV - A\|_1 \leq \alpha \cdot \text{OPT}$
with prob. 9/10

Assume : Exponential Time Hypothesis(ETH)
[Impagliazzo-Paturi-Zane'98]

Hardness - Under Exponential Time Hypothesis

ℓ_1 -Low Rank Approximation Hardness

Given : $A \in \mathbb{R}^{n \times n}, k \in \mathbb{N}, \alpha > 1$

Output : $U \in \mathbb{R}^{n \times k}, V \in \mathbb{R}^{k \times n}$ s.t. $\|UV - A\|_1 \leq \alpha \cdot \text{OPT}$
with prob. 9/10

Assume : Exponential Time Hypothesis(ETH)
[Impagliazzo-Paturi-Zane'98]

for arbitrarily small constant $\gamma > 0$

Hardness - Under Exponential Time Hypothesis

ℓ_1 -Low Rank Approximation Hardness

Given : $A \in \mathbb{R}^{n \times n}$, $k \in \mathbb{N}$, $\alpha > 1$

Output : $U \in \mathbb{R}^{n \times k}$, $V \in \mathbb{R}^{k \times n}$ s.t. $\|UV - A\|_1 \leq \alpha \cdot \text{OPT}$
with prob. 9/10

Assume : Exponential Time Hypothesis(ETH)
[Impagliazzo-Paturi-Zane'98]

for arbitrarily small constant $\gamma > 0$

for any algorithm running in $n^{O(1)}$ time

Hardness - Under Exponential Time Hypothesis

ℓ_1 -Low Rank Approximation Hardness

Given : $A \in \mathbb{R}^{n \times n}$, $k \in \mathbb{N}$, $\alpha > 1$

Output : $U \in \mathbb{R}^{n \times k}$, $V \in \mathbb{R}^{k \times n}$ s.t. $\|UV - A\|_1 \leq \alpha \cdot \text{OPT}$
with prob. 9/10

Assume : Exponential Time Hypothesis(ETH)
[Impagliazzo-Paturi-Zane'98]

for arbitrarily small constant $\gamma > 0$

for any algorithm running in $n^{O(1)}$ time

Requires : $\alpha > (1 + \frac{1}{\log^{1+\gamma} n})$

Hardness - Under Exponential Time Hypothesis

ℓ_1 -Low Rank Approximation Hardness

Given : $A \in \mathbb{R}^{n \times n}$, $k \in \mathbb{N}$, $\alpha > 1$

Output : $U \in \mathbb{R}^{n \times k}$, $V \in \mathbb{R}^{k \times n}$ s.t. $\|UV - A\|_1 \leq \alpha \cdot \text{OPT}$
with prob. 9/10

Assume : Exponential Time Hypothesis(ETH)
[Impagliazzo-Paturi-Zane'98]

for arbitrarily small constant $\gamma > 0$

for any algorithm running in $n^{O(1)}$ time

Requires : $\alpha > (1 + \frac{1}{\log^{1+\gamma} n})$

Implies a $2^{\Omega(1/\epsilon^{1-\gamma})}$ Hardness

Open Problems

- The hardness result is for $(1 + 1/\text{poly}(\log n))$ approximation, can we obtain a hardness result for $O(1)$ approximation ?

Open Problems

- The hardness result is for $(1 + 1/\text{poly}(\log n))$ approximation, can we obtain a hardness result for $O(1)$ approximation ?
- We have an $n^{\tilde{O}(k)}$ -time $\tilde{O}(k)$ -approximation algorithm, can either the approximation or the time be improved?

Open Problems

- The hardness result is for $(1 + 1/\text{poly}(\log n))$ approximation, can we obtain a hardness result for $O(1)$ approximation ?
- We have an $n^{\tilde{O}(k)}$ -time $\tilde{O}(k)$ -approximation algorithm, can either the approximation or the time be improved?
- We have a $\text{poly}(n)$ -time $O(\log n) \cdot \text{poly}(k)$ -approximation algorithm. Can the approximation be improved?

Conclusions

- Studied intractable matrix factorization problems through the lens of parameterized complexity

Conclusions

- Studied intractable matrix factorization problems through the lens of parameterized complexity
 - ▶ nonnegative matrix factorization

Conclusions

- Studied intractable matrix factorization problems through the lens of parameterized complexity
 - ▶ nonnegative matrix factorization
 - ▶ weighted low rank approximation

Conclusions

- Studied intractable matrix factorization problems through the lens of parameterized complexity
 - ▶ nonnegative matrix factorization
 - ▶ weighted low rank approximation
 - ▶ ℓ_1 -low rank approximation

Conclusions

- Studied intractable matrix factorization problems through the lens of parameterized complexity
 - ▶ nonnegative matrix factorization
 - ▶ weighted low rank approximation
 - ▶ ℓ_1 -low rank approximation
- Algorithm techniques

Conclusions

- Studied intractable matrix factorization problems through the lens of parameterized complexity
 - ▶ nonnegative matrix factorization
 - ▶ weighted low rank approximation
 - ▶ ℓ_1 -low rank approximation
- Algorithm techniques
 - ▶ polynomial system verifier

Conclusions

- Studied intractable matrix factorization problems through the lens of parameterized complexity
 - ▶ nonnegative matrix factorization
 - ▶ weighted low rank approximation
 - ▶ ℓ_1 -low rank approximation
- Algorithm techniques
 - ▶ polynomial system verifier
 - ▶ random matrices for dimensionality reduction to reduce the number of variables

Conclusions

- Studied intractable matrix factorization problems through the lens of parameterized complexity
 - ▶ nonnegative matrix factorization
 - ▶ weighted low rank approximation
 - ▶ ℓ_1 -low rank approximation
- Algorithm techniques
 - ▶ polynomial system verifier
 - ▶ random matrices for dimensionality reduction to reduce the number of variables
- Hardness techniques

Conclusions

- Studied intractable matrix factorization problems through the lens of parameterized complexity
 - ▶ nonnegative matrix factorization
 - ▶ weighted low rank approximation
 - ▶ ℓ_1 -low rank approximation
- Algorithm techniques
 - ▶ polynomial system verifier
 - ▶ random matrices for dimensionality reduction to reduce the number of variables
- Hardness techniques
 - ▶ random 4-SAT, ETH, etc.

Conclusions

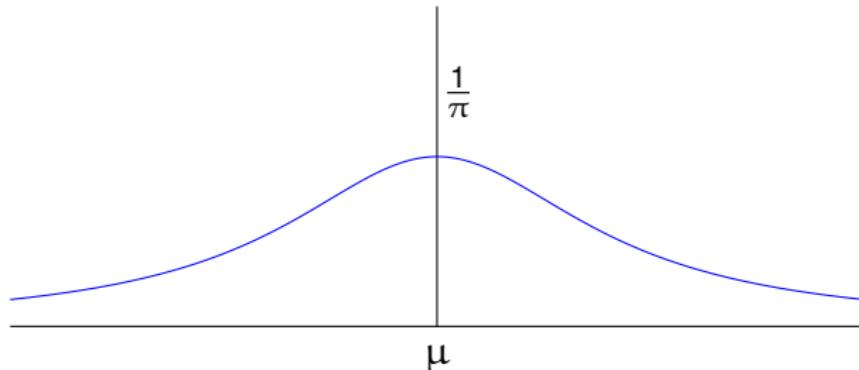
- Studied intractable matrix factorization problems through the lens of parameterized complexity
 - ▶ nonnegative matrix factorization
 - ▶ weighted low rank approximation
 - ▶ ℓ_1 -low rank approximation
- Algorithm techniques
 - ▶ polynomial system verifier
 - ▶ random matrices for dimensionality reduction to reduce the number of variables
- Hardness techniques
 - ▶ random 4-SAT, ETH, etc.
- Parameterized Complexity gives a way of coping with intractability for emerging machine learning problems

Thank you!

Questions?

Cauchy Distribution

Cauchy distribution $C(\mu, \gamma)$, pdf $f(x) = \frac{1}{\pi\gamma} \frac{\gamma^2}{(x-\mu)^2 + \gamma^2}$



For any three independent random variables

X, Y, Z drawn from Cauchy distribution.

For any $a, b \in \mathbb{R}$, $aX + bY \sim (|a| + |b|)Z$