Low-rank binary matrix approximation in column-sum norm

Fahad Panolan

Department of Computer Science and Engineering

Indian Institute of Technology Hyderabad, India

Joint work with Fedor V. Fomin, Petr Golovach, and Kirill Simonov.

Introduction

- Low-rank matrix approximation is the method of compressing a matrix by reducing its dimension.
- It is the basic component in data analysis including Principal Component Analysis (PCA) and has applications in machine learning, scientific computing, etc.

Low Rank Matrix Approximation

Given a matrix $\mathbf{A} \in \mathbb{R}^{m \times n}$ and a positive integer r, objective is a to find a matrix $\mathbf{B} \in \mathbb{R}^{m \times n}$ of rank at most r minimizing some matrix norm $||\mathbf{A} - \mathbf{B}||_{\nu}$.

Low Rank Matrix Approximation

Given a matrix $\mathbf{A} \in \mathbb{R}^{m \times n}$ and a positive integer r, objective is a to find a matrix $\mathbf{B} \in \mathbb{R}^{m \times n}$ of rank at most r minimizing some matrix norm $||\mathbf{A} - \mathbf{B}||_\nu$.

- Frobenius norm, $||\mathbf{A} - \mathbf{B}||_F^2 = \sum_{i,j}(a_{i,j} - b_{i,j})^2$.
- Spectral norm, $\|\mathbf{A} - \mathbf{B}\|_2 = \sup_{x \neq 0} \frac{\|(\mathbf{A}-\mathbf{B})\mathbf{x}\|_2}{\|\mathbf{x}\|_2}$.

Low Rank Matrix Approximation

Given a matrix $\mathbf{A} \in \mathbb{R}^{m \times n}$ and a positive integer r, objective is a to find a matrix $\mathbf{B} \in \mathbb{R}^{m \times n}$ of rank at most r minimizing some matrix norm $||\mathbf{A} - \mathbf{B}||_\nu$.

- Frobenius norm, $||\mathbf{A} - \mathbf{B}||_F^2 = \sum_{i,j}(a_{i,j} - b_{i,j})^2$.
- Spectral norm, $||\mathbf{A} - \mathbf{B}||_2 = \sup_{x \neq 0} \frac{||(\mathbf{A}-\mathbf{B})\mathbf{x}||_2}{||\mathbf{x}||_2}$.

Both the problems are solvable in polynomial time using Singular Value Decomposition.

Low Rank Matrix Approximation

Given a matrix $\mathbf{A} \in \mathbb{R}^{m \times n}$ and a positive integer r, objective is a to find a matrix $\mathbf{B} \in \mathbb{R}^{m \times n}$ of rank at most r minimizing some matrix norm $||\mathbf{A} - \mathbf{B}||_{\nu}$.

- Frobenius norm, $||\mathbf{A} - \mathbf{B}||_F^2 = \sum_{i,j}(a_{i,j} - b_{i,j})^2$.
- Spectral norm, $||\mathbf{A} - \mathbf{B}||_2 = \sup_{x \neq 0} \frac{||(\mathbf{A}-\mathbf{B})\mathbf{x}||_2}{||\mathbf{x}||_2}$.

Both the problems are solvable in polynomial time using Singular Value Decomposition.

We consider Low Rank Matrix Approximation of binary matrices over GF(2).

Low Rank Matrix Approximation Over GF(2)

$$A \quad \approx \quad C \quad \cdot \quad D \quad = \quad \mathrm{B}$$

$$\begin{bmatrix} 1 & 0 & 1 & \cdots & 0 \\ 0 & 0 & 0 & \cdots & 1 \\ 1 & 1 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 1 & 1 & \cdots & 1 \end{bmatrix}_{m \times n} \approx \begin{bmatrix} 1 & \cdots & 0 \\ 0 & \cdots & 1 \\ 1 & \cdots & 0 \\ \vdots & \vdots & \vdots \\ 0 & \cdots & 1 \end{bmatrix}_{m \times r} \begin{bmatrix} 1 & 1 & 0 & \cdots & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 1 & 0 & \cdots & 1 \end{bmatrix}_{r \times n}$$

Low Rank Matrix Approximation Over GF(2)

$$
\begin{array}{ccccc}
A & \approx & C & \cdot & D & = & \mathrm{B} \\
\begin{bmatrix} 1 & 0 & 1 & \cdots & 0 \\ 0 & 0 & 0 & \cdots & 1 \\ 1 & 1 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 1 & 1 & \cdots & 1 \end{bmatrix}_{m \times n} & \approx & \begin{bmatrix} 1 & \cdots & 0 \\ 0 & \cdots & 1 \\ 1 & \cdots & 0 \\ \vdots & \vdots & \vdots \\ 0 & \cdots & 1 \end{bmatrix}_{m \times r} & & \begin{bmatrix} 1 & 1 & 0 & \cdots & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 1 & 0 & \cdots & 1 \end{bmatrix}_{r \times n} & &
\end{array}
$$

$$\min_{\mathbf{B}} ||\mathbf{A} - \mathbf{B}||_{\nu}$$

Low Rank Matrix Approximation Over GF(2)

$$A \approx C \cdot D = \mathrm{B}$$

$$\begin{bmatrix} 1 & 0 & 1 & \cdots & 0 \\ 0 & 0 & 0 & \cdots & 1 \\ 1 & 1 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 1 & 1 & \cdots & 1 \end{bmatrix}_{m \times n} \approx \begin{bmatrix} 1 & \cdots & 0 \\ 0 & \cdots & 1 \\ 1 & \cdots & 0 \\ \vdots & \vdots & \vdots \\ 0 & \cdots & 1 \end{bmatrix}_{m \times r} \begin{bmatrix} 1 & 1 & 0 & \cdots & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 1 & 0 & \cdots & 1 \end{bmatrix}_{r \times n}$$

$$\min_{\mathbf{B}} ||\mathbf{A} - \mathbf{B}||_{\nu}$$

- Entry-wise ℓ_0-norm $\left(\sum_i \sum_j |a_{i,j} - b_{i,j}|\right)$: Low GF2 Rank Approximation

Low Rank Matrix Approximation Over GF(2)

$$A \approx C \cdot D = \mathrm{B}$$

$$\begin{bmatrix} 1 & 0 & 1 & \cdots & 0 \\ 0 & 0 & 0 & \cdots & 1 \\ 1 & 1 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 1 & 1 & \cdots & 1 \end{bmatrix}_{m \times n} \approx \begin{bmatrix} 1 & \cdots & 0 \\ 0 & \cdots & 1 \\ 1 & \cdots & 0 \\ \vdots & \vdots & \vdots \\ 0 & \cdots & 1 \end{bmatrix}_{m \times r} \begin{bmatrix} 1 & 1 & 0 & \cdots & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 1 & 0 & \cdots & 1 \end{bmatrix}_{r \times n}$$

$$\min_{\mathbf{B}} ||\mathbf{A} - \mathbf{B}||_{\nu}$$

- Entry-wise ℓ_0-norm $\left(\sum_i \sum_j |a_{i,j} - b_{i,j}|\right)$: Low GF2 Rank Approximation
- Column-sum norm $\left(||\mathbf{A} - \mathbf{B}||_1 = \max_j \sum_i |a_{i,j} - b_{i,j}|\right)$

ℓ_1-Rank-r Approximation over GF(2)

Known Results: Low GF2 Rank Approximation

Minimize entry-wise ℓ_0-norm, $\sum_i \sum_j |a_{i,j} - b_{i,j}|$

- NP-hard

[Gillis and Vavasis (2015)], [Dan, Hansen, Jiang, Wang, and Zhou (2015)]

- A PTAS with running time $n^{\mathcal{O}(\frac{2^{2r}}{\epsilon^2} \log \frac{1}{\epsilon})}$.

Known Results: Low GF2 Rank Approximation

Minimize entry-wise ℓ_0-norm, $\sum_i \sum_j |a_{i,j} - b_{i,j}|$

- NP-hard

[Gillis and Vavasis (2015)], [Dan, Hansen, Jiang, Wang, and Zhou (2015)]

- A PTAS with running time $n^{\mathcal{O}(\frac{2^{2r}}{\epsilon^2} \log \frac{1}{\epsilon})}$.
- A randomized linear time approximation scheme (Running time $f(k, \epsilon)nm$).

[Fomin, Golovach, Lokshtanov, Panolan, Saurabh (2019)]
[Ban, Bhattiprolu, Bringmann, Kolev, Woodruff (2019)]

Our Result: ℓ_1-Rank-r Approximation over GF(2)

Minimize the column-sum norm, $\max_j \sum_j |a_{i,j} - b_{i,j}|$

A simple reduction from Closest String shows that the problem is NP-hard.

Main result

For every $\varepsilon \in (0,1)$, there is a randomized $(1+\varepsilon)$-approximation algorithm of running time $m^{\mathcal{O}(1)} n^{\mathcal{O}(2^{4r} \cdot \varepsilon^{-4})}$.

Proof Overview

The proof has 3 steps.

Step 1:

Reduce the problem to BINARY CONSTRAINED k-CENTER (BCC)

Step 2:

Using dimension reduction technique reduce BCC to BINARY CONSTRAINED PARTITION CENTER (BCPC)

Step 3:

Solve BCPC using Integer Linear Programming

Step 1

$$\underbrace{\begin{pmatrix} a_1 & \cdots & a_n \end{pmatrix}}_{A} \approx \sum_{i=1}^{r} \underbrace{\begin{pmatrix} b_1^i & \cdots & b_n^i \end{pmatrix}}_{B_i} = \underbrace{\begin{pmatrix} b_1 & \cdots & b_n \end{pmatrix}}_{B}$$

Rank of $B_i = 1$ (i.e., all non-zero columns of B_i are identical).

Step 1

$$\underbrace{\begin{pmatrix} a_1 & \cdots & a_n \end{pmatrix}}_{A} \approx \sum_{i=1}^{r} \underbrace{\begin{pmatrix} b_1^i & \cdots & b_n^i \end{pmatrix}}_{B_i} = \underbrace{\begin{pmatrix} b_1 & \cdots & b_n \end{pmatrix}}_{B}$$

Rank of $B_i = 1$ (i.e., all non-zero columns of B_i are identical).

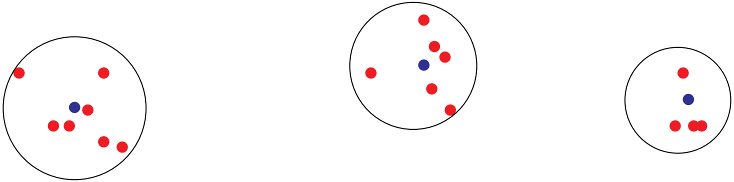
No. of distinct columns in B is at most 2^r.

Step 1

$$\underbrace{\begin{pmatrix} a_1 & \cdots & a_n \end{pmatrix}}_{A} \approx \sum_{i=1}^{r} \underbrace{\begin{pmatrix} b_1^i & \cdots & b_n^i \end{pmatrix}}_{B_i} = \underbrace{\begin{pmatrix} b_1 & \cdots & b_n \end{pmatrix}}_{B}$$

Rank of $B_i = 1$ (i.e., all non-zero columns of B_i are identical).

No. of distinct columns in B is at most 2^r.



- Minimize the maximum Hamming distance from red points to closest blue point.
- There is a relation between cluster centers.

Binary Constrained k-Center

- $R_1, \ldots, R_m$ – k-ary relations with elements from $\{0, 1\}$.
- $C = \{c_1, \ldots, c_k\} \subseteq \{0, 1\}^m$ satisfies $\mathcal{R} = \{R_1, \ldots, R_m\}$ if :

$$(c_1[i], \ldots, c_k[i]) \in R_i \text{ for all } i \in [m].$$

Binary Constrained k-Center

- $R_1, \ldots, R_m$ – k-ary relations with elements from $\{0, 1\}$.
- $C = \{c_1, \ldots, c_k\} \subseteq \{0, 1\}^m$ satisfies $\mathcal{R} = \{R_1, \ldots, R_m\}$ if :

$$(c_1[i], \ldots, c_k[i]) \in R_i \text{ for all } i \in [m].$$

Example:

- $m = 2$, $k = 3$
- $R_1 = \{(0, 0, 1), (1, 0, 0)\}$, and $R_2 = \{(1, 1, 1), (1, 0, 1), (0, 0, 1)\}$

Binary Constrained k-Center

- $R_1, \ldots, R_m$ – k-ary relations with elements from $\{0, 1\}$.
- $C = \{c_1, \ldots, c_k\} \subseteq \{0, 1\}^m$ satisfies $\mathcal{R} = \{R_1, \ldots, R_m\}$ if :

$$(c_1[i], \ldots, c_k[i]) \in R_i \text{ for all } i \in [m].$$

Example:

- $m = 2$, $k = 3$
- $R_1 = \{(0,0,1), (1,0,0)\}$, and $R_2 = \{(1,1,1), (1,0,1), (0,0,1)\}$
- $$c_1 = \begin{pmatrix} 0 \\ 1 \end{pmatrix}, c_2 = \begin{pmatrix} 0 \\ 0 \end{pmatrix}, c_3 = \begin{pmatrix} 1 \\ 1 \end{pmatrix} \qquad \text{satisfies } \mathcal{R} = \{R_1, R_2\}$$

Binary Constrained k-Center

- Input : $X \subseteq \{0,1\}^m$, $k \in \mathbb{N}$, and $\mathcal{R} = \{R_1, \ldots, R_m\}$.
- Output : Among all the sets $C = \{c_1, \ldots, c_k\}$ satisfying $\mathcal{R}$, find C

  minimizing $\max_{x \in X} d(x, C)$.

Reduction to BINARY CONSTRAINED k-CENTER (BCC)

$$A = \begin{pmatrix} a_1 & \cdots & a_n \end{pmatrix} \text{ and } r \in \mathbb{N}$$

- Partition columns of A into 2^r parts
- Find centers for each part s.t. centers are linear span of r vectors minimizing maximum distance distance of a column to the closest center.

Reduction to BINARY CONSTRAINED k-CENTER (BCC)

$$A = \begin{pmatrix} a_1 & \cdots & a_n \end{pmatrix} \text{ and } r \in \mathbb{N}$$

- Partition columns of A into 2^r parts
- Find centers for each part s.t. centers are linear span of r vectors minimizing maximum distance distance of a column to the closest center.

Reduction for r=2

- X is the the set of columns of A and for all $i \in [m]$, $R_i = \{(0,0,0,0), (0,0,1,1), (0,1,0,1), (0,1,1,0)\}$.

Step 2: BCC to BCPC

- Input : $X \subseteq \{0,1\}^m$, $k \in \mathbb{N}$, $\mathcal{R} = \{R_1, \dots, R_m\}$, and a partition $X_1 \uplus \dots \uplus X_k$ of X.
- Output : Among all the sets $C = \{c_1, \dots, c_k\}$ satisfying $\mathcal{R}$, find C minimizing

$$\max_{i=1}^{k} \max_{x \in X_i} d(x, c_i) = \mathsf{cost}(X_1 \uplus \dots \uplus X_k, C).$$

Step 2: Proof sketch

- **Dimension reduction [Ostrovsky and Rabani, 2002].** For any $Y \subseteq \{0,1\}^m$ of size $n+k$, and $\varepsilon > 0$, there is a linear map ψ from Y to $\{0,1\}^{\mathcal{O}(\frac{1}{\varepsilon^4}\log n)}$ such that the Hamming distances between any two vectors in $\psi(Y)$ are relatively preserved w.h.p.

There exits α s.t. $(1-\varepsilon)\alpha d(x,y) \leq d(\psi(x),\psi(y)) \leq (1+\varepsilon)\alpha d(x,y)$, $\forall x,y \in Y$

Step 2: Proof sketch

- **Dimension reduction [Ostrovsky and Rabani, 2002].** For any $Y \subseteq \{0,1\}^m$ of size $n+k$, and $\varepsilon > 0$, there is a linear map ψ from Y to $\{0,1\}^{\mathcal{O}(\frac{1}{\varepsilon^4}\log n)}$ such that the Hamming distances between any two vectors in $\psi(Y)$ are relatively preserved w.h.p.

  There exits α s.t. $(1-\varepsilon)\alpha d(x,y) \leq d(\psi(x),\psi(y)) \leq (1+\varepsilon)\alpha d(x,y)$, $\forall x,y \in Y$

- Let $Y = X \cup C$, where C is a hypothetical solution. Then, for any vector $x \in X$ and $c \in C$, the Hamming distance $d(\psi(x),\psi(c))$ is relatively preserved w.h.p.

Step 2: Proof sketch

- **Dimension reduction [Ostrovsky and Rabani, 2002].** For any $Y \subseteq \{0,1\}^m$ of size $n+k$, and $\varepsilon > 0$, there is a linear map ψ from Y to $\{0,1\}^{\mathcal{O}(\frac{1}{\varepsilon^4}\log n)}$ such that the Hamming distances between any two vectors in $\psi(Y)$ are relatively preserved w.h.p.

  There exits α s.t. $(1-\varepsilon)\alpha d(x,y) \leq d(\psi(x),\psi(y)) \leq (1+\varepsilon)\alpha d(x,y)$, $\forall x, y \in Y$

- Let $Y = X \cup C$, where C is a hypothetical solution. Then, for any vector $x \in X$ and $c \in C$, the Hamming distance $d(\psi(x),\psi(c))$ is relatively preserved w.h.p.

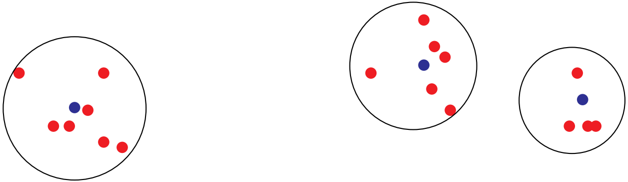
- Guess $\psi(C)$ and then partition X into k blocks based on distances between $\psi(X)$ and $\psi(C)$. Cost of this guess is $n^{\mathcal{O}(k/\varepsilon^4)}$.

Step 2: Proof sketch

- Let $Y = X \cup C$, where C is a hypothetical solution. Then, for any $x \in X$ and $c \in C$, the Hamming distance $d(\psi(x), \psi(c))$ is relatively preserved w.h.p.

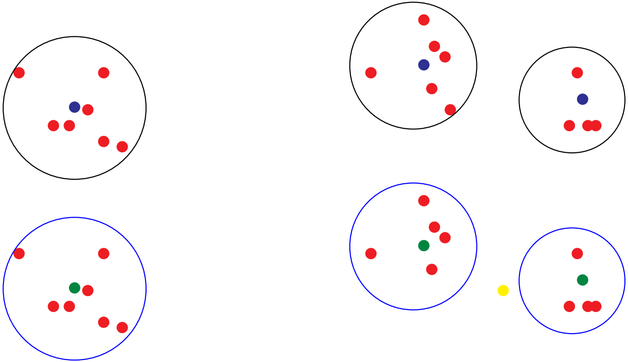
Step 2: Proof sketch

- Let $Y = X \cup C$, where C is a hypothetical solution. Then, for any $x \in X$ and $c \in C$, the Hamming distance $d(\psi(x), \psi(c))$ is relatively preserved w.h.p.



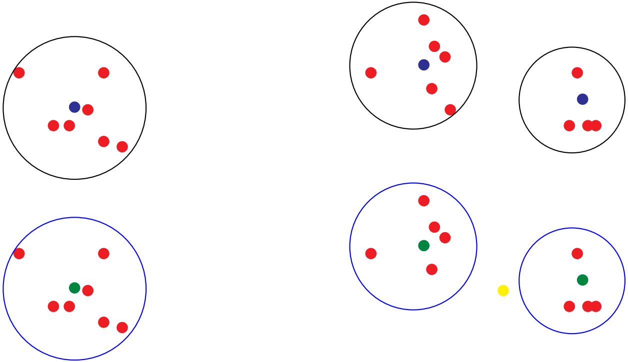
Step 2: Proof sketch

- Let $Y = X \cup C$, where C is a hypothetical solution. Then, for any $x \in X$ and $c \in C$, the Hamming distance $d(\psi(x), \psi(c))$ is relatively preserved w.h.p.



Step 2: Proof sketch

- Let $Y = X \cup C$, where C is a hypothetical solution. Then, for any $x \in X$ and $c \in C$, the Hamming distance $d(\psi(x), \psi(c))$ is relatively preserved w.h.p.



The above partition gives us an instance of BCPC. A $(1+\varepsilon)$-approx solution C' to the instance of BCPC is a $(1+\varepsilon)^2$-approx solution for the instance of BCC.

Step 3: Solving BCPC

- We formulate BCPC as Integer Programming formulation
- We solve it using randomized rounding method. This method is similar to the method used for PTAS of Closest String.

Step 3: Solving BCPC (Case 1: $OPT > m/c$ for a constant c)

- We formulate BCPC as Integer Programming formulation
- For every $j \in [m]$ and tuple $t \in R_j$, we have a variable $y_{j,t}$. Interpretation: if $y_{j,t} = 1$, then the row j of center vectors (solution) is same as tuple t.

Step 3: Solving BCPC (Case 1: $OPT > m/c$ for a constant c)

- We formulate BCPC as Integer Programming formulation
- For every $j \in [m]$ and tuple $t \in R_j$, we have a variable $y_{j,t}$. Interpretation: if $y_{j,t} = 1$, then the row j of center vectors (solution) is same as tuple t.

$$
\begin{aligned}
&\min d \\
&\text{subject to} \\
&\sum_{t \in R_j} y_{j,t} = 1, && \text{for all } j \in [m]; \\
&\sum_{j \in [m]} \sum_{t \in R_j} \chi_i(\mathbf{x}[j], t) \cdot y_{j,t} \leq d, && \text{for all } i \in [k] \text{ and } \mathbf{x} \in X_i \\
&y_{j,t} \in \{0, 1\}, && \text{for all } j \in [m] \text{ and } t \in R_j.
\end{aligned}
$$

Here, $\chi_i(\mathbf{x}[j], t) = 0$ if $\mathbf{x}[j] = t[i]$ and $\chi_i(\mathbf{x}[j], t) = 1$ if $\mathbf{x}[j] \neq t[i]$

Step 3: Solving BCPC (Case 1: $OPT > m/c$ for a constant c)

- Solve the LP relaxation of the above IP, and obtain fractional solution $y^\star_{j,t}$ ($j \in [m]$ and $t \in R_j$)
- Now, for each $j \in [m]$, independently with probability $y^\star_{j,t}$, we set $y'_{j,t} = 1$ and $y'_{j,t'} = 0$, for any $t' \in R_j \setminus \{t\}$.
- Then $y'_{j,t}$ ($j \in [m]$ and $t \in R_j$) form a solution to IP.
- Claim: $y'_{j,t}$ ($j \in [m]$ and $t \in R_j$) is a $(1+\varepsilon)$- approximate solution.

Step 3: Solving BCPC (Case 1: $OPT > m/c$ for a constant c)

$$\begin{aligned}
E[d(\mathbf{x}, \mathbf{c}_i)] &= E\left[\sum_{j\in[m]}\sum_{t\in R_j} \chi_i(\mathbf{x}[j], t) \cdot y'_{j,t}\right] \\
&= \sum_{j\in[m]}\sum_{t\in R_j} \chi_i(\mathbf{x}[j], t) \cdot E[y'_{j,t}] \\
&= \sum_{j\in[m]}\sum_{t\in R_j} \chi_i(\mathbf{x}[j], t) \cdot y^{\star}_{j,t} \leq d
\end{aligned}$$

Step 3: Solving BCPC (Case 1: $OPT > m/c$ for a constant c)

$$\begin{aligned} E[d(\mathbf{x}, \mathbf{c}_i)] &= E\left[\sum_{j \in [m]} \sum_{t \in R_j} \chi_i(\mathbf{x}[j], t) \cdot y'_{j,t}\right] \\ &= \sum_{j \in [m]} \sum_{t \in R_j} \chi_i(\mathbf{x}[j], t) \cdot E[y'_{j,t}] \\ &= \sum_{j \in [m]} \sum_{t \in R_j} \chi_i(\mathbf{x}[j], t) \cdot y^{\star}_{j,t} \leq d \end{aligned}$$

Fix $\delta = \frac{\varepsilon}{c}$. By Chernoff bound,

$$\Pr[d(\mathbf{x}, \mathbf{c}_i) > d + \varepsilon OPT] \leq \Pr[d(\mathbf{x}, \mathbf{c}_i) > d + \delta m] = \leq e^{-\frac{1}{3} m \delta^2}.$$

Step 3: Solving BCPC (Case 1: $OPT > m/c$ for a constant c)

$$
\begin{aligned}
E[d(\mathbf{x}, \mathbf{c}_i)] &= E\left[\sum_{j\in[m]} \sum_{t\in R_j} \chi_i(\mathbf{x}[j], t) \cdot y'_{j,t}\right] \\
&= \sum_{j\in[m]} \sum_{t\in R_j} \chi_i(\mathbf{x}[j], t) \cdot E[y'_{j,t}] \\
&= \sum_{j\in[m]} \sum_{t\in R_j} \chi_i(\mathbf{x}[j], t) \cdot y^{\star}_{j,t} \leq d
\end{aligned}
$$

Fix $\delta = \frac{\varepsilon}{c}$. By Chernoff bound,

$$\Pr[d(\mathbf{x}, \mathbf{c}_i) > d + \varepsilon OPT] \leq \Pr[d(\mathbf{x}, \mathbf{c}_i) > d + \delta m] =\leq e^{-\frac{1}{3} m \delta^2}.$$

Therefore, by the union bound,

$$\Pr[\text{There exist } i \in [k] \text{ and } \mathbf{x} \in X_i \text{ such that } d(\mathbf{x}, \mathbf{c}_i) > d + \varepsilon OPT] \leq n \cdot e^{-\frac{1}{3} m \delta^2} \leq n^{-2}$$

for $m = \Omega(\log n)$

Step 3: Solving BCPC (Case 1: $OPT \leq m/c$)

- We fix some co-ordinates of center vectors and then the remaining will have large optimum.
- That is, there exist $Y_1 \subseteq X_1, \ldots, Y_k \subseteq X_k$ such that $|Y_i| = O(1/\varepsilon)$ and fixing identical co-ordinates in Y_is will leads a good approximate solution.

Conclusion

- We gave PTAS for ℓ_1-RANK-r APPROXIMATION OVER GF(2). Running time is

$$m^{\mathcal{O}(1)} n^{\mathcal{O}(2^{4r} \cdot \varepsilon^{-4})}$$

Conclusion

- We gave PTAS for ℓ_1-Rank-r Approximation over GF(2). Running time is

$$m^{\mathcal{O}(1)} n^{\mathcal{O}(2^{4r} \cdot \varepsilon^{-4})}$$

- What about a EPTAS, i.e., with running time $f(\varepsilon)(mn)^{g(r)}$?

Conclusion

- We gave PTAS for ℓ_1-Rank-r Approximation over GF(2). Running time is
$$m^{\mathcal{O}(1)} n^{\mathcal{O}(2^{4r} \cdot \varepsilon^{-4})}$$

- What about a EPTAS, i.e., with running time $f(\varepsilon)(mn)^{g(r)}$?
An adaptation of a result on Closest String by Cygan, Lokshtanov, Pilipczuk, Pilipczuk, and Saurabh (2016) implies that such a result even for $r = 1$ is not possible assuming FPT$\neq$W[1].

Conclusion

- We gave PTAS for ℓ_1-Rank-r Approximation over GF(2). Running time is

$$m^{\mathcal{O}(1)} n^{\mathcal{O}(2^{4r} \cdot \varepsilon^{-4})}$$

- What about a EPTAS, i.e., with running time $f(\varepsilon)(mn)^{g(r)}$?
  An adaptation of a result on Closest String by Cygan, Lokshtanov, Pilipczuk, Pilipczuk, and Saurabh (2016) implies that such a result even for $r = 1$ is not possible assuming FPT$\neq$W[1].

Conclusion

- We gave PTAS for ℓ_1-Rank-r Approximation over GF(2). Running time is

$$m^{\mathcal{O}(1)} n^{\mathcal{O}(2^{4r} \cdot \varepsilon^{-4})}$$

- What about a EPTAS, i.e., with running time $f(\varepsilon)(mn)^{g(r)}$?
  An adaptation of a result on Closest String by Cygan, Lokshtanov, Pilipczuk, Pilipczuk, and Saurabh (2016) implies that such a result even for $r = 1$ is not possible assuming FPT$\neq$W[1].

- Also, existence of an algorithm with running time $f(\varepsilon)n^{o(1/\varepsilon)}$ even for $r = 1$ will contradict Exponential Time Hypothesis (ETH).

Conclusion

- We gave PTAS for ℓ_1-Rank-r Approximation over GF(2). Running time is

$$m^{\mathcal{O}(1)} n^{\mathcal{O}(2^{4r}\cdot\varepsilon^{-4})}$$

- What about a EPTAS, i.e., with running time $f(\varepsilon)(mn)^{g(r)}$?
  An adaptation of a result on Closest String by Cygan, Lokshtanov, Pilipczuk, Pilipczuk, and Saurabh (2016) implies that such a result even for $r = 1$ is not possible assuming FPT≠W[1].

- Also, existence of an algorithm with running time $f(\varepsilon)n^{o(1/\varepsilon)}$ even for $r = 1$ will contradict Exponential Time Hypothesis (ETH).

- But, algorithm with running $f(r,\varepsilon)(mn)^{poly(1/\varepsilon)}$ is open.

Thank You.