# Faster Algorithms for Binary Matrix Factorization

Ravi Kumar, Rina Panigrahy, Ali Rahimi, David P. Woodruff
Google and CMU

# Binary Matrix Factorization

- Given A in $\{0,1\}^{m \times n}$, find $U \in \{0,1\}^{m \times k}$ and $V \in \{0,1\}^{k \times n}$ to minimize

$$|U \cdot V - A|_p^p$$

- For an m x n matrix C, $|C|_p^p = \sum_{i,j} |C_{i,j}|^p$

- $U \cdot V$ can be
  - Integer product: $\langle U_{i*}, V_{*j} \rangle = \sum_{\ell=1,\ldots,k} U_{i,\ell} \cdot V_{\ell,j}$
  - Mod 2 product: $\langle U_{i*}, V_{*j} \rangle = \sum_{\ell=1,\ldots,k} U_{i,\ell} \cdot V_{\ell,j}$ mod 2
  - Boolean product: $\langle U_{i*}, V_{*j} \rangle = \vee \left(_{\ell=1,\ldots,k} U_{i,\ell} \wedge V_{\ell,j}\right)$

# Approximation Algorithms

- All variants are NP-hard for any p-norm

- What about randomized O(1)-approximation algorithms?

- Output $U \in \{0,1\}^{m \times k}$ and $V \in \{0,1\}^{k \times n}$ for which
$$|U \cdot V - A|_p^p \leq O(1) \cdot \min_{U' \in \{0,1\}^{m \times k},\, V' \in \{0,1\}^{k \times n}} |U' \cdot V' - A|_p^p$$

- $f(k) \cdot \mathrm{poly}(mn)$ randomized O(1)-approximation algorithms
  - $f(k) = 2^{2^{\Theta(k)}}$ [BBBKLW, FGLPS]
  - Doubly-exponential running time is prohibitive

# Complexity Analysis

- A in $\{0,1\}^{m \times n}$ is a bipartite incidence matrix
  - $A_{i,j} = 1$ iff i-th left vertex incident to j-th right vertex

- If $U \cdot V$ is Boolean product, the 1-entries of $U \cdot V$ are the edges in a union of k bipartite cliques (“bicliques”)
  - the i-th biclique has left vertex set support($U^i$) and right vertex set support($V^i$)

- Under the Exponential Time Hypothesis (ETH), $2^{2^k}$ time is needed to decide if biclique covering number is k
  - Rules out $2^{2^{o(k)}}$ time for any multiplicative approximation and for any p norm

- What about integer product and mod 2 product?

# Integer Product

- $2^{2^k}$ time lower bound does not apply to integer product!
- If $U \cdot V = A$ for integer product $U \cdot V$, the 1-entries of $U \cdot V$ are the edges in a multiset union of k bicliques
  - If $U \cdot V = A$, the biclique partition number is k
- Can decide if biclique partition number is at most k in $2^{O(k^2)}$ time [CIK]
- What if we only know $U \cdot V \approx A$ for $U \in \{0,1\}^{m \times k}$ and $V \in \{0,1\}^{k \times n}$?
  - To find O(1)-approximate U and V, previous algorithms take $2^{2^k}$ or $\min(m,n)^{k^{O(1)}}$ time
  - p = 1 minimizes edges in symmetric difference between input and multiset union of bicliques
- *Can we obtain fast O(1)-approximation algorithms for integer product?*

# OLED Motivation for Integer Product

- A display can be viewed as an m x n matrix of pixels

- Passive displays render one row at a time
  - human eye integrates this into an image
  - brightness inversely proportional to number of rows
  - active displays are brighter because they add memory to keep the pixel illuminated for duration of the image, but they are expensive

- We observe that rendering a row has same cost as rendering a rank-1 image
  - brightness proportional to duration of rendering, which is rank of decomposition
  - binary factors allow us to use cheap voltage drivers on rows and columns

# Our Result

- For any $p \geq 1$, there's a $2^{(k^{\lceil \frac{p}{2} \rceil + 1}) \log k} \text{poly}(mn)$ time algorithm outputting $U \in \{0,1\}^{m \times k}$ and $V \in \{0,1\}^{k \times n}$ with

$$|U \cdot V - A|_p^p \leq O(1) \cdot \min_{U' \in \{0,1\}^{m \text{ x } k},\, V' \in \{0,1\}^{k \times n}} |U' \cdot V' - A|_p^p,$$

where $U \cdot V$ is integer product, i.e., $\langle U_{i*}, V_{*j} \rangle = \sum_{\ell = 1, \ldots, k} U_{i,\ell} \cdot V_{\ell, j}$

- Assuming ETH, there's a $2^{\Omega(k)} \text{poly}(mn)$ time lower bound

# Our Techniques

- For a subset S of rows of matrix C, let $S \cdot C$ be the matrix consisting of the rows in S
- Let $U^* \in \{0,1\}^{n \times k}$, $V^* \in \{0,1\}^{k \times n}$ be the minimizer to $|U^* V^* - A|_p^p$
- Theorem: Let s = $k^{\left\lceil \frac{p}{2} \right\rceil + 1} \log k$. There is a subset $S \cdot A$ of s rows of A, and an s x s diagonal matrix D with entries in {1, 2, 4, 8, …, ns}, with

$$\forall V \in R^{k \times n}, \qquad |D \cdot S \cdot U^* \cdot V - D \cdot S \cdot A|_p^p = \Theta(1) \cdot |U^* V - A|_p^p$$

- Proof: properties of Lewis weights ("optimized $l_p$-leverage scores") and triangle inequality

# Interpreting the Theorem

- Theorem: Let s = $k^{\lceil \frac{p}{2} \rceil + 1} \log k$. There is a subset $S \cdot A$ of s rows of A, and an s x s diagonal matrix D with entries in {1, 2, 4, 8, …, ns}, with $\forall V \in \{0,1\}^{k \times n}$, $|D \cdot S \cdot U^* \cdot V - D \cdot S \cdot A|_p^p = \Theta(1) \cdot |U^* V - A|_p^p$

---

- If we had $D \cdot S \cdot U^*$ and $D \cdot S \cdot A$, can solve for each column of V separately in $2^k \cdot \text{poly}(sk)$ time by guessing all $2^k$ possibilities and choosing the best one
- Given V, we can then solve for each row of U separately in $2^k \cdot \text{poly}(sk)$ time, where the i-th row $U_i$ is the minimizer to $|U_i \cdot V - A|_p^p$. Overall, we'd get $\Theta(1)$-approximation
- But we don't know $D \cdot S \cdot U^*$ and $D \cdot S \cdot A$

# Guess a Sketch Framework [RSW]

- Theorem: Let s = $k^{\left\lceil \frac{p}{2} \right\rceil + 1} \log k$. There is a subset $S \cdot A$ of s rows of A, and an s x s diagonal matrix D with entries in {1, 2, 4, 8, …, ns}, with
  $$\forall V \in R^{k \times n}, \qquad |D \cdot S \cdot U^* V - D \cdot S \cdot A|_p^p = \Theta(1) \cdot |U^* V - A|_p^p$$

- $S \cdot U^*$ is binary and $s \times k$ => only $2^{sk}$ possibilities
- D is s x s diagonal s x s with entries in {1, 2, 4, 8, …, ns} => only $(\log(ns))^s$ possibilities
- Try all $S \cdot U^*$ and D => only $(\log(ns))^s \cdot 2^{sk} \leq 2^{O(sk)} \text{poly}(n)$ possibilities

- But $S \cdot A$ can be an arbitrary s x n binary matrix, too many possibilities

# Preconditioning via Clustering

- If A had only $2^k$ distinct rows, then there are only $2^{sk}$ possibilities for $S \cdot A$, and only $(\log n)^s \, 2^{sk} \leq 2^{O(sk)} \mathrm{poly}(n)$ possibilities for $D \cdot S \cdot A$

- [CGTS] Given a set P of n points in $R^d$, there is an algorithm running in poly(nd) time which outputs $(C_1, \ldots, C_{2^k})$ and $\left(c_1, \ldots, c_{2^k}\right)$, with $c_i \in P$, and

$$\sum_{i=1,\ldots,2^k} \sum_{x \in C_i} |x - c_i|_p^p \leq \kappa_p \cdot \mathrm{OPT}_{2^k}$$

where $\kappa_p$ depends only on p, and $\mathrm{OPT}_{2^k}$ is the optimal $2^k$-clustering cost

- Let B be the m x n matrix whose i-th row is the nearest center $c_j$ to $A_i$
- B is binary and has only $2^k$ distinct rows. Replace A with B!

# Putting it All Together

- Let $U' \cdot V'$ be an O(1)-approximate binary low rank approximation to B
- Let $U^* \cdot V^*$ be an optimal binary low rank approximation to A
- Let OPT be the optimal binary low rank approximation cost to A

- $$\begin{aligned} |A - U' \cdot V'|_p &\le |A - B|_p + |B - U' \cdot V'|_p \\ &\le |A - B|_p + |B - U^* \cdot V^*|_p \\ &\le |A - B|_p + |B - A|_p + |A - U^* \cdot V^*|_p \\ &= 2|A - B|_p + \mathrm{OPT} \end{aligned}$$

- $|A - B|_p$ = O(1) OPT, since any binary low rank matrix has $\le 2^k$ distinct rows

# Conclusions

- For any $p \geq 1$, there's a $2^{(k^{\lceil \frac{p}{2} \rceil + 1}) \log k}$ $\mathrm{poly}(mn)$ time algorithm outputting $U \in \{0,1\}^{m \times k}$ and $V \in \{0,1\}^{k \times n}$ for which

$$|U \cdot V - A|_p^p \leq O(1) \cdot \min_{U\prime \in \{0,1\}^{m \text{ x } k},\, V\prime \in \{0,1\}^{k \times n}} |U' \cdot V' - A|_p^p$$

- When $U \cdot V$ is mod 2 product, we show a $2^{O(k^3)} \mathrm{poly}(mn)$ time algorithm outputting $U \in \{0,1\}^{m \times r}$ and $V \in \{0,1\}^{r \times n}$ with

$$|U \cdot V - A|_p^p \leq O(1) \cdot \min_{U\prime \in \{0,1\}^{m \text{ x } k},\, V\prime \in \{0,1\}^{k \times n}} |U' \cdot V' - A|_p^p$$

where r = O(k log m). Since $U \cdot V$ is binary, error measure doesn't depend on p

- Empirically, we find clustering into k groups instead of $2^k$ already gives good binary low rank approximations