Faster Algorithms for Binary Matrix Factorization

Ravi Kumar [1] **Rina Panigrahy** [1] **Ali Rahimi** [1] **David P. Woodruff** [2]

Abstract

We give faster approximation algorithms for well-studied variants of Binary Matrix Factorization (BMF), where we are given a binary $m \times n$ matrix A and would like to find binary rank-k matrices U, V to minimize the Frobenius norm of $U \cdot V - A$.

In the first setting, $U \cdot V$ denotes multiplication over $\mathbb{Z}$, and we give a constant-factor approximation algorithm that runs in $2^{O(k^2 \log k)}\text{poly}(mn)$ time, improving upon the previous $\min(2^{2^k}, 2^n)\text{poly}(mn)$ time. Our techniques generalize to minimizing $\|U \cdot V - A\|_p$ for $p \geq 1$, in $2^{O(k^{\lceil p/2 \rceil + 1} \log k)}\text{poly}(mn)$ time. For $p = 1$, this has a graph-theoretic consequence, namely, a $2^{O(k^2)}\text{poly}(mn)$-time algorithm to approximate a graph as a union of disjoint bicliques. In the second setting, $U \cdot V$ is over $\text{GF}(2)$, and we give a bicriteria constant-factor approximation algorithm that runs in $2^{O(k^3)}\text{poly}(mn)$ time to find binary rank-$O(k \log m)$ matrices U, V whose cost is as good as the best rank-k approximation, improving upon $\min(2^{2^k} mn, \min(m, n)^{k^{O(1)}}\text{poly}(mn))$ time.

1. Introduction

In the low rank approximation problem, we are given an $m \times n$ matrix A and would like to approximate A as $U \cdot V$, where U is $m \times k$ and V is $k \times n$. Here k is the rank parameter, which is typically a small integer. Approximating A by $U \cdot V$ has a number of advantages, e.g., it takes only $(m+n)k$ parameters to store U and V versus mn parameters to store the original A, and for an arbitrary n-dimensional column vector x, one can compute $U \cdot Vx$ in $(m + n)k$ time as opposed to mn time to compute Ax. A natural notion of approximation is Frobenius norm error, where one seeks to find U and V so as to minimize $\|A - UV\|_F^2$; the latter is defined to be $\sum_{i=1}^m \sum_{j=1}^n (A_{i,j} - \langle U_{i,*}, V_{*,j} \rangle)^2$, where $U_{i,*}$ denotes the ith row of U and $V_{*,j}$ the jth column of V. There are a number of other notions of error studied, such as entrywise-ℓ_1 error $\|A - UV\|_1 = \sum_{i=1}^m \sum_{j=1}^n |A_{i,j} - \langle U_{i,*}, V_{*,j} \rangle|$, studied in Song et al. (2017). More generally, entrywise-ℓ_p error $\|A - UV\|_p^p = \sum_{i=1}^m \sum_{j=1}^n |A_{i,j} - \langle U_{i,*}, V_{*,j} \rangle|^p$ and other M-Estimator loss functions have been recently studied (Chierichetti et al., 2017; Song et al., 2018).

In many applications data is binary or categorical rather than real-valued. For example, in the UCI repository, nearly half of the data sets are categorical. In the *Binary Matrix Factorization* (BMF) problem, the input matrix $A \in \{0,1\}^{m \times n}$ is binary, and we would like to factorize it into binary matrices $U \in \{0,1\}^{m \times k}$, $V \in \{0,1\}^{k \times n}$. There are many formulations of this problem, depending on what the inner product $\langle U_{i,*}, V_{*,j} \rangle$ should mean, and we will focus on two variants.

The first is the standard inner product over the integers. For this notion, it is helpful to think of A as being the incidence matrix of a bipartite graph with m vertices on the left, n vertices on the right, and an edge existing from the jth left vertex to the j'th right vertex if and only if $A_{j,j'} = 1$. Then, noting that $U \cdot V = \sum_{i=1}^k U_{*,i} V_{i,*}$ is the sum of k rank-1 matrices, each being the outer product of a column of U and the corresponding row of V, the entries of $U_{*,i} V_{i,*}$ are exactly the edges in a bipartite clique (biclique) between the vertices j on the left for which $U_{i,j} = 1$ and the vertices j' on the right for which $V_{j',i} = 1$. Thus, in this problem we seek to represent A as a multi-set union of bicliques. Determining the minimal k for which there is zero error is the *Bipartite Clique Partition* problem, studied in Orlin (1977); Fleischner et al. (2007); Chalermsook et al. (2014); Chandran et al. (2016); Neumann (2018). We also present a novel application of this notion of inner product to OLED displays, which is a central motivation for this work.

Another natural notion of inner product is when $\langle U_{i,*}, V_{*,j} \rangle$ is taken over the binary field $\text{GF}(2)$, where arithmetic operations are defined modulo 2. This latter model has been applied to Independent Component Analysis (ICA) over string data, and has attracted attention from the signal processing community (Yeredor, 2011; Gutch et al., 2012; Painsky et al., 2015). It serves as an important tool in dimension

[1]Google, 1600 Amphitheater Parkway, Mountain View, CA, US. [2]CMU, 5000 Forbes Ave, Pittsburgh, PA, US. Part of this work was done while the author was visiting Google, and part while visiting the Simons Institute for the Theory of Computing. Correspondence to: David Woodruff <dwoodruf@cs.cmu.edu>.

Proceedings of the 36^{th} International Conference on Machine Learning, Long Beach, California, PMLR 97, 2019.

reduction for high-dimensional data with binary attributes (Koyutürk & Grama, 2003; Shen et al., 2009; Jiang et al., 2014). There are also numerous heuristics proposed for this problem (Shen et al., 2009; Fu et al., 2010; Jiang et al., 2014; Koyutürk & Grama, 2003). The model is also studied in column subset selection (Dan et al., 2015).

There is yet another model known as the *Boolean model*, or *Boolean Factor Analysis*, for which $\langle U_{i,*}, V_{*,j}\rangle = \vee_\ell (U_{i,\ell} \wedge V_{\ell,j})$, i.e., the OR of ANDs of the corresponding entries of $U_{i,*}$ and $V_{*,j}$. This has found applications in data mining such as latent variable analysis, topic models, association rule mining, and database tiling (Seppänen et al., 2003; Šingliar & Hauskrecht, 2006; Belohlavek & Vychodil, 2010). This also appears under other names in the literature, such as the Discrete Basis Problem (Miettinen et al., 2006) or Minimal Noise Role Mining Problem (Vaidya et al., 2007; Lu et al., 2012; Mitra et al., 2016).

In recent independent work of Ban et al. (2019) and Fomin et al. (2018), a generic randomized algorithm was proposed for solving BMF under many different notions of inner product, including $\mathrm{GF}(2)$ low rank approximation and Boolean Factor Analysis. For the Bipartite Clique Partition problem, these results only apply to minimizing $\|U \cdot V - A\|_0$, where for a matrix B, $\|B\|_0$ denotes the number of non-zero entries of B. Ban et al. (2019) show that one can obtain a constant-factor approximation to BMF in $2^{2^{O(k)}} mn^{1+o(1)}$ time, while Fomin et al. (2018) give an improved running time of $2^{2^{O(k)}} mn$. Both works observe that the *Bipartite Clique Covering* problem coincides with the Boolean Factor Analysis problem. Moreover, Chandran et al. (2016) show this problem requires $\min(2^{2^{\Omega(k)}}, 2^{n^{\Omega(1)}})$ time under the Exponential Time Hypothesis (ETH), a standard complexity assumption. Consequently, if one considers algorithms for classes of BMF that includes Boolean Factor Analysis, one needs to spend $\min(2^{2^{\Omega(k)}}, 2^{n^{\Omega(1)}})$ time.

While these results are tight for Boolean Factor Analysis, they leave open the possibility of doing much better for the other well-studied variants of BMF, such as the Bipartite Clique Partition or $\mathrm{GF}(2)$ low-rank approximation problems. The doubly-exponential running time required of the algorithms in Ban et al. (2019); Fomin et al. (2018) can be quite restrictive. Moreover, it is unclear how to improve the running time of these algorithms in practice, as the doubly exponential times come from guessing and enumerating all possibilities of 2^k samples in an unknown optimal clustering.

1.1. Our contributions

We give the first constant-factor approximation algorithms for BMF, where the inner product is the standard inner product over the integers, i.e., the Bipartite Clique Partition problem, in *singly-exponential* time. More precisely, for a certain absolute constant $C > 1$, we show how to find $U \in \{0,1\}^{m\times k}$ and $V \in \{0,1\}^{k\times n}$ for which

$$\|U\cdot V - A\|_p^p \le C \min_{U'\in\{0,1\}^{m\times k}, V'\in\{0,1\}^{k\times n}} \|U'\cdot V' - A\|_p^p.$$

Our algorithm runs in $2^{O(k^2\log k)}\mathrm{poly}(mn)$ time for $p \in [1,2]$, and in $2^{O(k^{\lceil p/2\rceil+1}\log k)}\mathrm{poly}(mn)$ time for $p > 2$.

It was known how to solve the *exact* version of this problem where $A = U \cdot V$ in $2^{O(k^2)} \cdot \mathrm{poly}(mn)$ time (Chandran et al., 2016). But nothing was known about the approximate version of this problem for any error measure. Here we obtain $2^{\mathrm{poly}(k)}\mathrm{poly}(mn)$ time algorithms with entry-wise ℓ_p-error for any constant $p \ge 1$, significantly generalizing algorithms for the exact version. Moreover, since the exact version of the problem is a special case of the approximate version with relative error, and any exact algorithm requires $2^{\Omega(k)}$ time (Chandran et al., 2016) assuming ETH, it follows that our algorithms are optimal up to the precise $\mathrm{poly}(k)$ factor in the exponent assuming ETH.

For $p = 1$, our error measure has a natural combinatorial interpretation. Namely, suppose we are given an unweighted bipartite graph G with m vertices on the left and n vertices on the right. Suppose we wish to approximate G by the multiset union H of k bicliques so as to minimize the number of edges in the symmetric difference, i.e., to find H with

$$|E(G)\Delta E(H)| \le C \min_{\substack{\text{a disjoint union}\\ H' \text{ of } k \text{ cliques}}} |E(G)\Delta E(H')|,$$

where $C > 1$ is a constant approximation factor. While there are algorithms to decide if G can be expressed as the disjoint union of k bicliques in $2^{O(k^2)} \cdot \mathrm{poly}(mn)$ time, surprisingly, no algorithms were known for its approximate version. We give the first constant-factor approximation algorithm that runs in $2^{O(k^2\log k)}\mathrm{poly}(mn)$ time.

We also give the fastest known bicriteria constant-factor approximation when the multiplication $U \cdot V$ is over the finite field $\mathrm{GF}(2)$. We achieve $2^{O(k^3)}\mathrm{poly}(mn)$ time to output binary rank $O(k\log m)$ matrices U and V whose cost is as good as the best rank-k approximation. This improves the previous $\min(2^{2^k}mn, \min(m,n)^{O(\mathrm{poly}(k))}\mathrm{poly}(mn))$ time. Here the first term in the minimum follows from both Ban et al. (2019) and Fomin et al. (2018), while the second term follows from Ban et al. (2019). We note that the algorithms of Ban et al. (2019) and Fomin et al. (2018) are able to provide $(1+\epsilon)$-approximations, whereas our algorithms only provide fixed constant-factor approximations. Also, in the case of $\mathrm{GF}(2)$ low-rank approximation, our algorithms are bicriteria. However, we stress that our algorithms are exponentially faster than theirs, making them polynomial time even for k as large as $\frac{\sqrt{\log n}}{\log\log n}$ for $p \in [1,2]$ and $\log^{\Omega(1)} n$ for constant $p > 2$. In contrast, their algorithms are already super-polynomial time for any $k = \omega(\log\log n)$.

1.2. A motivation and an application

The motivation behind this work was a driving scheme to make *passive* displays brighter so they can compete against their more expensive *active* counterparts. Passive OLED displays render an image by illuminating one row at a time. The rows are illuminated in rapid succession, and the human eye integrates this sequence into an image. The apparent brightness of an image is therefore inversely proportional to the number of rows in passive OLED. To make the image brighter, *active* displays add a memory layer to each pixel allowing them to stay illuminated for the duration of the image so that apparent brightness does not depend on the height of the display. But this additional layer dramatically increases cost. We have observed that passive displays have the often neglected electrical ability to illuminate many rows simultaneously, as long as the image being shown is a rank-1 matrix. Row-by-row rendering is just one way to represent an image as the sum of rank-1 matrices, each rank-1 image having at most one non-zero row. The apparent brightness of a given pixel is proportional to the amount of time that pixel is illuminated under this sequence, which is inversely proportional to the rank of the decomposition. BMF seeks a lower rank decomposition than the row-by-row decomposition, therefore pixels are illuminated for longer time in the sequence shown to the viewer. The binary constraints on the decomposition allows us to use simple voltage drivers on the rows and columns of the display instead of an expensive bank of video-rate digital to analog-to-digital converters.

2. Technical overview

All the algorithms we offer adhere to the following schema:

Input: $A \in \{0,1\}^{m\times n}$ and an integer k (the desired rank of the decomposition).
Output: $U \in \{0,1\}^{m\times k}$, $V \in \{0,1\}^{k\times n}$ that approximately minimize $\|A - U \cdot V\|$; the specifics differ depending on the norm $\|\cdot\|$ and whether $\cdot$ is over integers or $\mathrm{GF}(2)$.

1. Compute $I(A)$, a matrix that approximates A using 2^k distinct rows.
2. Guess a binary sketching matrix S and the matrix SU^* (The matrix U^* is the ideal minimizer of the factorization problem to solve.)
3. Compute V', the minimizer of $\|SU^*V - SI(A)\|$ by enumerating its columns one at a time.
4. Repeat steps 2 and 3 with different guesses to find the best V'.
5. Compute U', the minimizer of $\|UV' - A\|$ by enumerating its rows one at a time.
6. Output (U', V').

Here is a simple idea that almost works, inspired by Razenshteyn et al. (2016); Ban et al. (2019). It is known that there is a distribution on $O(k) \times m$ matrices S such that with constant probability, (i) for any $m \times k$ matrix U, $\|SUx\|_2 = \Theta(\|Ux\|_2)$ simultaneously for all vectors x, and (ii) for any fixed $m \times n$ matrix A, $\|SA\|_F = \Theta(1)\|A\|_F$. Consider the hypothetical optimization problem $\min_{V\in\{0,1\}^{k\times n}} \|U^*V - A\|_F$, where (U^*, V^*) is the minimizer of $\|U \cdot V - A\|_F^2$. We cannot quite solve this optimization problem since we do not know U^*, but we can replace it with the hypothetical optimization problem $\min_{V\in\{0,1\}^{k\times n}} \|SU^*V - SA\|_F$. To solve this problem, in addition to drawing S, one could "guess" SU^*, after which we can solve for each column of V' independently in 2^k time by enumerating all possibilities of the column and computing the one with minimal cost. Having found V', we can then solve the problem $\min_{U\in\{0,1\}^{m\times k}} \|UV' - A\|_F$ by solving for each row of U independently in 2^k time, giving $2^k m$ total time. The problem with this simple scheme is that the entries of SU^* can be $O(\log m)$ bit integers even if U^* is a binary matrix, which means there would be $m^{\Omega(k^2)}$ matrices SU^* to enumerate. This already exceeds the $2^{O(k^2 \log k)}$ time bound we desire.

We consider a refinement where we draw matrices S of dimension $O(k \log k) \times m$ that enjoy properties (i) and (ii) above, with the key difference that S samples a subset of $O(k \log k)$ rows of U^* according to their leverage scores, and rescales these rows. We write $S = DT$, where T is a row selection matrix, and D is a diagonal matrix which w.l.o.g. can be assumed to have all entries being powers of 2 between 1 and $\mathrm{poly}(m)$. There are only $2^{O(k^2 \log k)}$ possibilities of $T \cdot U^*$ since U^* is itself binary. Moreover, one can guess D in $(\log^{O(k \log k)} m)$-time and this amount of time turns out to be less than $2^{O(k^2 \log k)}\mathrm{poly}(mn)$.

We have explained how to guess SU^* without sampling all possible S, but to solve for V', we also need to estimate SA. SA only has $O(k \log k)$-dimensional columns, and therefore there are only $2^{O(k \log k)}$ possibilities for each column of TA. However, TA has n columns, and so enumerating all TA would require $2^{O(nk \log k)}$ guesses. Alternatively, since T samples $O(k \log k)$ rows and there are only m possible rows, we could enumerate all $m^{O(k \log k)}$ rows to sample to form both TU^* and TA, at which point guessing D would be inexpensive. However, this $m^{O(k \log k)}$ time is still prohibitive.

A critical observation now is that if A only had 2^k *distinct rows*, there would only be $2^{O(k^2 \log k)}$ possibilities for the matrix TA. Since there are only $2^{O(k^2 \log k)}$ possibilities for U^* and $(\log^{O(k \log k)} m)$ possibilities for D, we could then formulate each possibility of the optimization problem $\min_{V\in\{0,1\}^{k\times n}} \|SU^*V - SA\|_F$, solve each one in $O(2^k n)$ time by solving for each column of V by trying all 2^k possibilities, and choose the best solution we found. In general, of course, A does not not have 2^k dis-

tinct rows. So our scheme considers the m rows of A as points in $\mathbb{R}^n$ and runs a constant-factor 2^k-means approximation algorithm on these points; such algorithms run in $2^{O(k)}\mathrm{poly}(mn)$ time. Suppose the means are $d_1, \ldots, d_{2^k}$, and let $I(A)$ be an $m \times n$ cluster indicator matrix, i.e., if the center d_i serves the jth row A_j of A, then the jth row of $I(A)$ is equal to d_i. Moreover, one can assume that $I(A)$ is in fact binary by replacing each d_i with the closest row of A; this only changes the cost by a constant factor by the triangle inequality. Notice that for any binary matrix V, $\|U^*V - A\|_F^2$ is at least the 2^k-means cost of A since U^*V has at most 2^k distinct rows since it is a rank-k binary matrix. Hence, if we instead solve the optimization problem $\min_{V \in \{0,1\}^{k\times n}} \|U^*V - I(A)\|_F$, and then back-solve for U as before, then by applying the triangle inequality we will have that the $U'V'$ we find will be an $O(1)$-approximation to the original matrix A. But now $I(A)$ is a binary matrix with only 2^k distinct rows, and we can do this efficiently!

This schema can approximately minimize $\|U \cdot V - A\|$ up to a constant factor, where the matrix product $U \cdot V$ is performed over the integers. In that situation, S samples $O(k \log k)$ rows. To compute the matrix $I(A)$, we run a 2^k-means algorithm on the rows of A to identify 2^k distinct rows of A, and replace each row of A with one of the means thus identified.

This same schema can minimize $\|U \cdot V - A\|_p^p$. In that case, S selects rows using the so-called ℓ_p-Lewis weights (Cohen & Peng, 2015) instead of the leverage scores to form an $O(k^{\lceil p/2 \rceil \log k}) \times m$ sampling matrix. Moreover, one can replace the 2^k-means algorithm with a constant-factor approximation algorithm for finding 2^k centers so as to minimize sums of pth powers of ℓ_p-distances. Such an algorithm is implied by the results in Charikar et al. (2002) for the 2^k-median problem, since they only need an approximate triangle inequality, which holds for pth powers of distances.

The schema also applies to minimizing $\|U \cdot V - A\|_F$ when the product $U \cdot V$ is matrix multiplication over $\mathrm{GF}(2)$. To compute $I(A)$, we use an $O(1)$-approximate 2^k-means clustering algorithm. Note that since the entries of $U \cdot V$ and A are now binary, $\|U \cdot V - I(A)\|_F^2$ is the number of disagreements of $U \cdot V$ and $I(A)$. Thus it suffices to use an ℓ_0-sketch S from the streaming literature; however for us it is important for the multiplication $S \cdot U$ to be *performed over* $\mathrm{GF}(2)$ so that we have associativity, namely, that $S(U \cdot V) = (SU) \cdot V$. Unfortunately this does not seem to be possible, but what is instead possible is to write S as a sequence of $1 + \log_2 m$ matrices $S^0, S^1, \ldots, S^{\log_2 m} \in \{0,1\}^{O(k)\times m}$ so that the sketched optimization problem now becomes $\frac{1}{k}\sum_{i=0}^{\log_2 m} 2^i \|S^i \cdot (UV - I(A))\|_2^2$, where now each multiplication by S^i is indeed over $\mathrm{GF}(2)$. Note that we crucially need the integer weights 2^i to be outside of the matrix product, as there is no $\mathrm{GF}(2)$ analog of them. While this works and gives the first $2^{\mathrm{poly}(k)}\mathrm{poly}(mn)$-time algorithm for obtaining a constant-factor approximation to this problem, the weights ultimately cause our output to be a rank-$O(k \log m)$ rather than a rank-k approximation.

3. BMF with Frobenius norm error

We first show how our techniques can find $U \in \{0,1\}^{m\times k}$ and $V \in \{0,1\}^{k\times n}$ for which

$$\|U \cdot V - A\|_F \leq C \cdot \min_{U' \in \{0,1\}^{m\times k}, V' \in \{0,1\}^{k\times n}} \|U' \cdot V' - A\|_F,$$

where $C > 1$ is a constant, and for a matrix $B \in \mathbb{R}^{m\times n}$, $\|B\|_F = \left(\sum_{i=1}^n \sum_{j=1}^n B_{i,j}^2\right)^{1/2}$ is its Frobenius norm. We call this the *k-BMF with Frobenius norm error*.

Our algorithm uses an approximation algorithm for the k-means problem as a black box. For a given set of points $P \subset \mathbb{R}^d$, in the *k-means problem*, the goal is to find a partition of P into k clusters $(C_1, \ldots, C_k)$ with corresponding centers $(c_1, \ldots, c_k)$ that minimize the sum of the squared distances of all points in P to their corresponding center, i.e.,

$$\operatorname*{argmin}_{(C_1,\ldots,C_k),(c_1,\ldots,c_k)} \sum_{i=1}^{k} \sum_{x \in C_i} \|x - c_i\|^2,$$

where $\|\cdot\|$ is the Euclidean distance; let OPT_k be the mininum value.

Theorem 1 (Kanungo et al. (2004)) *Given a set P of n points in $\mathbb{R}^d$, for any k and any constant $\epsilon > 0$, there is a randomized algorithm running in $\mathrm{poly}(nd)$ time which, with probability $1 - \frac{1}{mn}$, outputs $(C_1, \ldots, C_k)$, and $(c_1, \ldots, c_k)$ for which*

$$\sum_{i=1} \sum_{x \in C_i} \|x - c_i\|^2 \leq (9 + \epsilon)\mathrm{OPT}_k.$$

Suppose we are given an instance $A \in \{0,1\}^{m\times n}$ of the k-BMF problem with Frobenius norm error. We first run the algorithm of Theorem 1 on the rows of A with parameter 2^k and constant $\epsilon > 0$. I.e., we treat the m rows of A as our pointset P of m points in $\mathbb{R}^n$. By Theorem 1, we output $(C_1, \ldots, C_{2^k})$ and $(c_1, \ldots, c_{2^k})$ that is a $(9+\epsilon)$-approximation to OPT_{2^k}. Note that the centers $c_1, \ldots, c_{2^k}$ need not be binary, so we first transform them. For each C_i, let d_i be the point x in C_i for which $\|x - c_i\|$ is minimized. We need the following inequality for squared distances:

Fact 2 *For every three points $x, y, z \in \mathbb{R}^n$,*

$$\|x - z\|^2 \leq 2(\|x - y\|^2 + \|y - z\|^2).$$

Using Fact 2, the choice of d_i, and the guarantee on c_i,

$$\sum_{i=1}^{2^k}\sum_{x\in C_i}\|x-d_i\|^2 \le 2\sum_{i=1}^{2^k}\sum_{x\in C_i}(\|x-c_i\|^2+\|c_i-d_i\|^2)$$
$$\le \quad 4\sum_{i=2}^{2^k}\sum_{x\in C_i}\|x-c_i\|^2 \quad\le\quad (36+O(\epsilon))\mathrm{OPT}_{2^k}.$$

The row cluster indicator matrix $I(A)$ can be constructed as follows: for each rows of A, if the row is in C_i, then replace it with the point d_i. Notice that

$$\|I(A)-A\|_F^2 \le (36+O(\epsilon))\mathrm{OPT}_{2^k}, \tag{1}$$

Like $I(A)$, for any matrices $U \in \{0,1\}^{m\times k}$ and $V \in \{0,1\}^{k\times n}$, the matrix $U\cdot V$ has at most 2^k distinct rows. Consequently,

$$\|U\cdot V-A\|_F^2 \ge \mathrm{OPT}_{2^k}. \tag{2}$$

Now suppose for a value $C \ge 1$, we could find $U \in \{0,1\}^{m\times k}$ and $V\in\{0,1\}^{k\times n}$ for which

$$\|U\cdot V-I(A)\|_F^2 \le C\cdot \min_{U',V'}\|U'\cdot V'-I(A)\|_F^2, \tag{3}$$

where $U'\in\{0,1\}^{m\times k}, V'\in\{0,1\}^{k\times n}$ here and below. Then for this choice of U and V we would have:

$$\begin{aligned}
\|U\cdot V-A\|_F^2 &\overset{\text{Fact 2}}{\le} 2(\|U\cdot V-I(A)\|_F^2+\|I(A)-A\|_F^2)\\
&\overset{(3)}{\le} 2(C\min_{U',V'}\|U'\cdot V'-I(A)\|_F^2+\|I(A)-A\|_F^2)\\
&\overset{\text{Fact 2}}{\le} 2(C\min_{U',V'}2(\|U'\cdot V'-A\|_F^2+\|A-I(A)\|_F^2))\\
&\quad +2\|I(A)-A\|_F^2\\
&= 4C\min_{U',V'}\|U'\cdot V'-A\|_F^2+(4C+2)\|A-I(A)\|_F^2\\
&\overset{(1)}{\le} 4C\min_{U',V'}\|U'\cdot V'-A\|_F^2\\
&\quad +(4C+2)(36+O(\epsilon))\mathrm{OPT}_{2^k}\\
&\overset{(2)}{\le} (4C+(4C+2)(36+O(\epsilon)))\min_{U',V'}\|U'\cdot V'-A\|_F^2.
\end{aligned}$$

Hence, for constant C and ϵ, this particular U and V would provide a constant-factor approximation.

It remains to find U and V satisfying (3) for a constant $C\ge 1$. To do so, we use the following result on leverage score sampling.

Theorem 3 (e.g., Woodruff (2014)) *For any $m\times k$ matrix U and $m\times n$ matrix B, there exists a subset T of $r = O(k\log k)$ rows and an $r\times r$ diagonal matrix D with entries between 1 and $O(m)$ for which*

1. *Simultaneously for all vectors $x\in\mathbb{R}^k$,*
$$\|DU_Tx\|_2^2 = (1\pm\tfrac{1}{2})\cdot\|Ux\|_2^2,$$
2. $\|DB_T\|_F^2 \le 4\|B\|_F^2$,
3. *The entries of D are powers of 2 and between 1 and* $\mathrm{poly}(m)$.

Here for a matrix C, C_T denotes the $r\times k$ submatrix of C consisting of the rows in T.

Remark 4 The bounds in Theorem 3 can be deduced from standard properties of leverage scores. It is known that if one samples $O(k\log k)$ rows of U according to the leverage scores of U, and forms a sampling and rescaling matrix D, where the jth row of D is equal to $1/\sqrt{p_i}$ if we sample row i with probability p_i in the jth repetition, then the first property holds. In fact, such probabilities can be rounded to powers of 2 since it suffices for the probabilities to be overestimates to the actual values, and this rounding at most doubles the size of T. Also, with high probability we do not choose any i for which $p_i \le 1/\Theta(mk\log k)$, implying the third property. For the second property, it suffices to observe that for any matrix B, $E[\|DB_T\|_F^2] = \|B\|_F^2$, and then apply a Markov bound.

There are at most $2^{O(k^2\log k)}$ distinct subsets of $O(k\log k)$ rows of $I(A)$, so we try all such subsets T of rows. We also try all possibilities of the corresponding D, and by Remark 4, there are only $O(\log m)^{O(k\log k)}$ total guesses, which is $2^{O(k\log k\log\log m)}$. If $\log m \le 2^k$, this is still $2^{O(k^2\log k)}$ time; otherwise $\log m > 2^k$, and so $O(\log m)^{O(k\log k)} = 2^{\log^2\log m\log\log\log m} \le m$. Consequently, there are at most $2^{O(k^2\log k)}\mathrm{poly}(mn)$ unique combinations of T and D.

For each guess of T and D, we next guess DU_T^*. Since U^* is a binary matrix, and we know D, this is just $2^{O(k^2\log k)}$ guesses. We know $DI(A)_T$, and so can solve for $\mathrm{argmin}_V\|DU_T^*V-DI(A)_T\|_F^2$. To do so, we can solve for each column of V independently, in 2^k time, by trying all possibilities. Thus, the total time is $2^k\mathrm{poly}(mn)$. Given V, we then solve $\mathrm{argmin}_U\|UV-I(A)\|_F^2$, which we can again do by solving for each row of U independently. The total time is $2^k\mathrm{poly}(mn)$. We output the U,V minimizing $\|UV-I(A)\|_F^2$ over all possible guesses that we find.

Consider the right guess, so that $\|DU_T^*y\|_2^2 = (1\pm 1/2)\|U^*y\|_2^2$ for all vectors y, and also, by Theorem 3, we can assume that for the matrix $B = U^*V^*-I(A)$, we have $\|DB_T\|_F\le 2\|B\|_F$. Here U^*V^* is the optimal solution.

The cost of the U and V that we find is upper bounded by the cost for the right guess of D and S, and in this case is:

$$\|UV-I(A)\|_F \quad\le\quad \|U^*V-I(A)\|_F$$

$$
\begin{aligned}
&\leq \|U^*V - U^*V^*\|_F + \|U^*V^* - I(A)\|_F \\
&\leq (3/2)\|DU_T^*V - DU_S^*V^*\|_F + \|U^*V^* - I(A)\|_F \\
&\leq (3/2)\|DU_T^*V - DI(A)_T\|_F \\
&\quad +(3/2)\|DI(A)_T - DU_T^*V^*\|_F + \|U^*V^* - I(A)\|_F \\
&\leq (3/2)\|DU_T^*V^* - DI(A)_S\|_F \\
&\quad +(3/2)\|DU_T^*V^* - DI(A)_S\|_F + \|U^*V^* - I(A)\|_F \\
&\leq 2\cdot(3/2)\|U^*V^* - I(A)\|_F + 4\|U^*V^* - I(A)\|_F \\
&= 7\|U^*V^* - I(A)\|_F,
\end{aligned}
$$

where the first inequality follows since our choice of U was optimal for the given V that we found, the second inequality is the triangle inequality, the third inequality follows from Property 1 of Theorem 3, the fourth inequality is the triangle inequality, the fifth inequality follows from our choice of V which was optimal with respect to DU_T^* and $DI(A)_T$, and the sixth inequality follows from Property 2 of Theorem 3. Thus, we have found U and V satisfying (3) for a constant $C \geq 1$, which completes the proof. We summarize our results with the following theorem.

Theorem 5 *There is a constant $C \geq 1$, and an algorithm running in $2^{O(k^2\log k)}\mathrm{poly}(mn)$ time, which given an $m \times n$ binary matrix A, finds $U \in \{0,1\}^{m\times k}$ and $V \in \{0,1\}^{k\times n}$ for which $\|UV - A\|_F^2 \leq C \cdot \min_{U'\in\{0,1\}^{m\times k}, V'\in\{0,1\}^{k\times n}} \|U'V' - A\|_F^2$.*

4. Generalization to p-norm error

We next show how to solve the p-norm version of k-BMF, where we seek $U \in \{0,1\}^{m\times k}$ and $V \in \{0,1\}^{k\times n}$ so that

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

where $C > 1$ is an absolute constant, and for a matrix $B \in \mathbb{R}^{m\times n}$, $\|B\|_p^p = \sum_{i=1}^n \sum_{j=1}^n |B_{i,j}|^p$ is its entrywise p-norm. For $p = 1$, this coincides with covering a bipartite graph with bicliques to minimize the symmetric difference in the multiset union of the edge sets.

To compute $I(A)$ we use approximation algorithms for the *metric k-median problem* as a black box, appropriately generalized to sums of pth powers of distances. The goal of this problem is to partition a set of points $P \subset \mathbb{R}^d$ into k clusters $(C_1, \ldots, C_k)$ with corresponding centers $(c_1, \ldots, c_k)$ that minimize the sum of distances of all points in P to their corresponding center, namely, the quantity

$$\operatorname*{argmin}_{(C_1,\ldots,C_k),(c_1,\ldots,c_k)} \sum_{i=1}^k \sum_{x\in C_i} \|x - c_i\|_p^p,$$

where $\|x - c_i\|_p^p = \sum_{j=1}^d |x_j - c_{i,j}|^p$.

There are polynomial time approximation algorithms for this problem for general metrics, and the following suffices.

Theorem 6 (Charikar et al. (2002)) *Given a set P of n points in $\mathbb{R}^d$, for any value of k, there is an algorithm running in $\mathrm{poly}(nd)$ time which outputs $(C_1, \ldots, C_k)$ and $(c_1, \ldots, c_k)$ for which*

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

where $\kappa_p \geq 1$ is an absolute constant depending only on p.

The algorithm and the proof follow mostly along the lines of the Frobenius norm version but now we use Theorem 6 instead of k-means and Lewis weight sampling (Cohen & Peng, 2015) instead of leverage score sampling. We leave the details to the Supplementary Material.

Theorem 7 *There is a constant $C \geq 1$, and an algorithm running in $2^{O(k^{\lceil p/2\rceil+1}\log k)}\mathrm{poly}(mn)$ time, which given an $m \times n$ binary matrix A, finds $U \in \{0,1\}^{m\times k}$ and $V \in \{0,1\}^{k\times n}$ for which $\|UV - A\|_p^p \leq C \cdot \min_{U'\in\{0,1\}^{m\times k}, V'\in\{0,1\}^{k\times n}} \|U'V' - A\|_p^p$.*

5. BMF over $\mathrm{GF}(2)$

For multiplication over $\mathrm{GF}(2)$, our algorithm is bicriteria. It outputs U, V of rank $O(k\log m)$ in $2^{O(k^3)}\mathrm{poly}(mn)$ time, and achieves a constant-factor approximation compared to the best rank-k approximation. This improves previous $\min(2^{2^k}, \min(m,n)^k)\mathrm{poly}(mn)$ time for bicriteria algorithms.

We again first run the algorithm of Theorem 1 to find the matrix $I(A)$. The algorithm does not depend on the notion of multiplication, since any $U \cdot V$ will still have at most 2^k distinct rows.

After finding $I(A)$, as before it suffices to find a constant $C \geq 1$ and $U \in \{0,1\}^{m\times k}$, $V \in \{0,1\}^{k\times n}$ for which (3) holds. The main difference is that we can no longer use Theorem 3, and instead need the following result. Note that the first property is *lopsided*, i.e., we can only guarantee we do not "shrink" vectors, though we may "dilate" them:

Theorem 8 *There is a distribution on $1 + \log_2 m$ matrices $S^0, S^1, \ldots, S^{\log_2 m}$, each in $\{0,1\}^{O(k)\times m}$, with the following properties: with probability at least $9/10$ over the choice of $S^0, S^1, \ldots, S^{\log_2 m}$, for any matrix $U \in \{0,1\}^{m\times k}$ and matrix $B \in \{0,1\}^{m\times n}$,*

1. *Simultaneously for all vectors $x \in \{0,1\}^k$, $\frac{1}{k}\sum_{i=0}^{\log_2 m} 2^i \|S^i \cdot Ux\|_2^2 \geq \frac{1}{200}\cdot\|Ux\|_2^2$, and*
2. *$\frac{1}{k}\sum_{i=0}^{\log_2 m} \|S^iB\|_F^2 \leq 100\|B\|_F^2$,*

where matrix multiplications are over $\mathrm{GF}(2)$.

Proof: The proof is inspired from estimating algorithms for ℓ_0, for example Kane et al. (2010). Choose each row of each S^i independently from the following distribution: in-

dependently include each coordinate of $\{1, 2, \ldots, m\}$ with probability 2^{-i}, and on each included coordinate, choose it to be independent and uniformly random in $\{0, 1\}$. Non-included coordinates are set to 0.

Consider a vector of the form $y = Ux$. Let $Y = \text{supp}(y)$ be the set of coordinates of y which are equal to 1. Consider a row z of S^i and let $Z = \text{supp}(z)$. Conditioned on $|Y \cap Z| > 0$, the probability that $\langle z, y\rangle = 1$ is exactly $1/2$, where multiplication is over GF(2). If $|Y \cap Z| = 0$, then this probability is 0. Hence,

$$\mathbf{E}[\langle z, y\rangle] \quad = \quad \tfrac{1}{2}\Pr[|Y \cap Z| > 0] = \tfrac{1}{2}\left(1 - (1 - 2^{-i})^{|Y|}\right).$$

On the one hand, we have $\left(1 - (1 - 2^{-i})^{|Y|}\right) \geq 1 - e^{-|Y|/2^i} \geq |Y|/2^i - |Y|^2/2^{2i+1}$.

Suppose that $2^i \leq |Y|$. By a Chernoff bound, we have that with probability $1 - 2^{-2k}$, $\|S^i y\|_2^2 \geq k/100$, where we have chosen the constant factor in the $O(k)$ number of rows of S^i to be sufficiently large. Letting i^* be maximal for which $2^{i*} \leq |Y|$, we have

$$\begin{aligned}\frac{1}{k}\sum_{i=0}^{\log_2 m} 2^i \|S^i \cdot Ux\|_2^2 \quad &\geq \quad 2^{i^*}\frac{k}{100} \geq \frac{1}{100}2^{i^*} \\ &\geq \quad \frac{1}{200}|Y| = \frac{1}{200}\|Ux\|_2^2,\end{aligned}$$

and now the first part of the theorem follows by a union bound over the 2^k different possible values of x.

Before proving the second part of the theorem, we first fix a vector $y = Ux$, and bound $\mathbf{E}[\frac{1}{k}\sum_{i=i^*+1}^{\log_2 m} 2^i \|S^i \cdot Ux\|_2^2]$. Consider an i for which $2^i > |Y|$. We have $\left(1 - (1 - 2^{-i})^{|Y|}\right) \leq |Y|/2^i$. Hence,

$$\mathbf{E}[\frac{1}{k}\sum_{i=i^*+1}^{\log_2 m} 2^i \|S^i \cdot Ux\|_2^2] \leq \frac{1}{k}\sum_{i=i^*+1}^{\log_2 m} 2^i \cdot \frac{|Y|}{2^i} \leq 2\|Ux\|_2^2.$$

Note also that $\frac{1}{k}\sum_{i=0}^{i^*} 2^i \|S^i \cdot Ux\|_2^2 \leq \frac{2}{k}2^{i^*} \cdot k \leq 2\|Ux\|_2^2$, and so this bound also holds in expectation. Consequently, $\mathbf{E}[\frac{1}{k}\sum_{i=i^*+1}^{\log_2 m} 2^i \|S^i \cdot Ux\|_2^2] \leq 4\|Ux\|_2^2$.

Returning to the second part of the theorem, it is enough to apply this expectation bound, linearity of expectation across the n columns of B, and a Markov bound to conclude that also with probability $1 - 1/25$, $\frac{1}{k}\sum_{i=0}^{\log_2 m} \|S^i B\|_F^2 \leq 100\|B\|_F^2$.

By a union bound, both parts of the theorem hold simultaneously with probability at least $9/10$. ■

Details appear in the Supplementary Material.

Theorem 9 *There is a constant $C \geq 1$, and an algorithm running in $2^{O(k^3)}\text{poly}(mn)$ time and succeeding with probability $1 - 1/(mn)$, which given an $m \times n$ binary matrix A, finds $U \in \{0,1\}^{m \times O(k \log m)}$ and $V \in \{0,1\}^{O(k \log m) \times n}$ for which $\|UV - A\|_F^2 \leq C \cdot \min_{U' \in \{0,1\}^{m \times k}, V' \in \{0,1\}^{k \times n}} \|U'V' - A\|_F^2$, where all multiplications are performed over* GF(2).

6. Experiments

In this section we present simple experimental results for our algorithm for k-BMF in the Frobenius norm error. The purpose of these experiments is to demonstrate the practical applicability of our algorithm to real-world data, especially for images, which primarily motivated our work.

The datasets we use are the standard MNIST database of handwritten digits (`yann.lecun.com/exdb/mnist/`) and the ORL databases of black-and-white faces (`www.cl.cam.ac.uk/research/dtg/attarchive/facedatabase.html`). For MNIST, we consider 60,000 images in the training set. Each image is 28×28 and we normalize it to be binary, i.e., each entry in the matrix is in $\{0, 1\}$. For ORL, we consider the 400 images in the dataset, where each image is 112×92; since the images are grayscale, we threshold the matrix to be binary using a value of 0.33. Our goal is to run various algorithms for k-BMF on each matrix and measure the Frobenius norm and the time taken to compute the factorization, for various values of k.

Baseline and algorithm. For the baseline, we use an implementation of an algorithm by Zhang et al. (2007). This algorithm works by extending the standard Non-negative Matrix Factorization (NMF) to the BMF case. The algorithm works in an alternating minimization style and uses gradient descent to compute the optimum in each step of the iteration. By its very iterative nature, the factors obtained by this algorithm are guaranteed to be binary only in the limit. We run this baseline for 10,000 iterations and round the entries of the factors to be in $\{0, 1\}$. (The results are mostly unchanged even without this rounding.) We use an implementation of this algorithm that is available as part of the Python package `pymf` (`github.com/ChrisSchinnerl/pymf3`).

For our algorithm, we use the k-means based approach described in Section 3. However, since our goal is to get a practical algorithm, we essentially stop at (1), i.e., we do not use enumeration and leverage score sampling to find the best factorization. We apply k-means to the rows of the matrix and replace each non-center row by its closest center. As we argued in Section 3, this step already has provable (but weak) guarantees. For solving k-means we use the vanilla k-means, available as part of `scikit-learn` (`scikit-learn.org`) in Python. Since this implementation can return Steiner points, we use nearest-neighbors to replace each Steiner point by its closest input point.

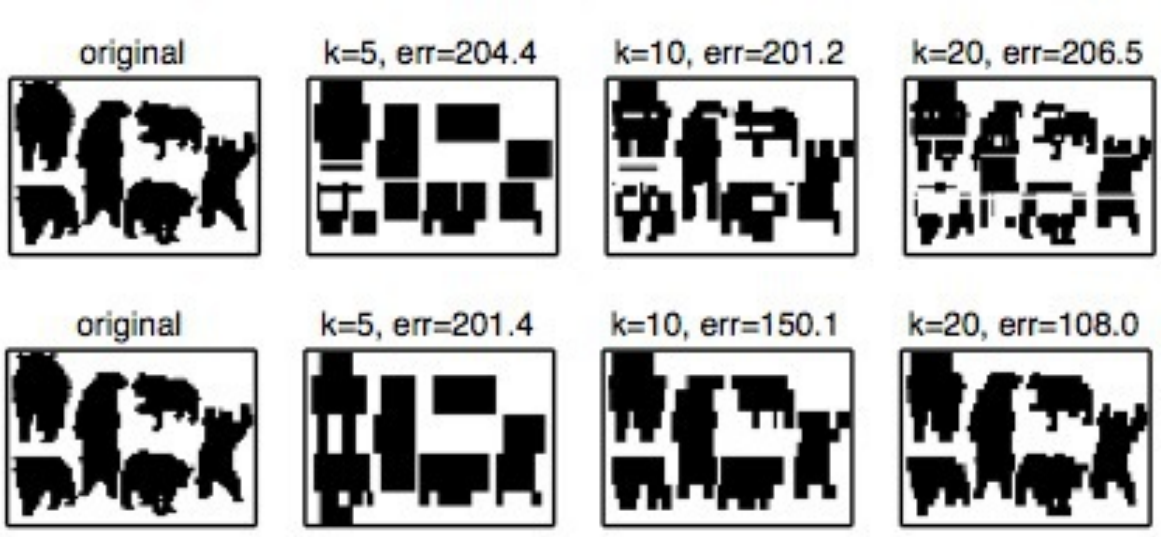

Figure 1. Example output of the baseline and our algorithm on a 700×490 black-and-white image. The top row shows the output for various values of k for the baseline; the bottom row shows the output of our algorithm. The Frobenius norm error is also shown for each k.

In the experiments, our goal is to measure the average Frobenius error and the running time, as a function of k.

k	Baseline		Our algorithm	
	Error	Time (ms)	Error	Time (ms)
	MNIST			
2	9.52	30.05	8.03	24.99
3	8.53	31.26	6.75	29.47
5	9.41	36.26	5.03	39.03
10	14.59	44.18	2.52	68.02
	ORL			
5	39.54	67.49	27.28	73.99
10	54.08	93.73	22.17	115.82
20	66.84	140.46	17.22	204.54
30	71.83	284.28	13.85	307.49
50	76.58	412.92	8.22	500.76

Table 1. Results of our algorithm on MNIST and ORL compared to the baseline (Zhang et al., 2007), as function of k. Here, the error is the average Frobenius norm error and the time is the average time taken to compute the factorization.

Results. Table 1 shows the results of the baseline and our algorithm for the MNIST and ORL. As we see, our algorithm obtains much smaller Frobenius norm error compared to the baseline, especially for large k. It is interesting to note the "non-monotone" behavior of the NMF-based baseline; this seems to have been the case even in some earlier work. In terms of running times, our algorithm is only marginally worse compared to the baseline even for large k.

Figure 1 shows the sample output of the baseline and our algorithm for an image, for different k. The non-monotone behavior of the baseline (with respect to k) is clear. In addition to having a smaller error, our algorithm also produces an image that is visually more faithful to the original.

7. Conclusions

In this paper we studied the classic BMF problem for different error measures. We obtained a new, faster, algorithm with provable performance guarantees, improving upon the previous algorithms substantially. In fact, we introduced a general program that let us obtain a family of results on this topic, including ones on approximately decomposing a bipartite graph into bicliques and approximating BMF over the finite field $\mathrm{GF}(2)$. Our experiments indicate that a practical version of our algorithm performs better on black-and-white images when compared to recent but more heuristic NMF-based approaches to BMF. These heuristics lack any performance guarantees whereas our algorithms, including the practical versions, have approximation guarantees.

Interesting future work includes further improving our algorithms from a practical viewpoint and making the leverage sampling and enumeration steps efficient. Using our approximate biclique algorithm to find near-dense communities in bipartite graphs is also a promising research direction.

References

Ban, F., Bhattiprolu, V., Bringmann, K., Kolev, P., Lee, E., and Woodruff, D. P. A PTAS for ℓ_p-low rank approximation. In *SODA*, pp. 747–766, 2019.

Belohlavek, R. and Vychodil, V. Discovery of optimal factors in binary data via a novel method of matrix decomposition. *JCSS*, 76(1):3–20, 2010.

Chalermsook, P., Heydrich, S., Holm, E., and Karrenbauer, A. Nearly tight approximability results for minimum biclique cover and partition. In *ESA*, pp. 235–246, 2014.

Chandran, L. S., Issac, D., and Karrenbauer, A. On the parameterized complexity of biclique cover and partition. In *IPEC*, pp. 11:1–11:13, 2016.

Charikar, M., Guha, S., Tardos, É., and Shmoys, D. B. A constant-factor approximation algorithm for the k-median problem. *JCSS*, 65(1):129–149, 2002.

Chierichetti, F., Gollapudi, S., Kumar, R., Lattanzi, S., Panigrahy, R., and Woodruff, D. P. Algorithms for ℓ_p low-rank approximation. In *ICML*, pp. 806–814, 2017.

Cohen, M. B. and Peng, R. l_p row sampling by Lewis weights. In *STOC*, pp. 183–192, 2015.

Dan, C., Hansen, K. A., Jiang, H., Wang, L., and Zhou, Y. On low rank approximation of binary matrices. *CoRR*, abs/1511.01699, 2015.

Fleischner, H., Mujuni, E., Paulusma, D., and Szeider, S. Covering graphs with few complete bipartite subgraphs. In *FSTTCS*, pp. 340–351, 2007.

Fomin, F. V., Golovach, P. A., Lokshtanov, D., Panolan, F., and Saurabh, S. Approximation schemes for low-rank binary matrix approximation problems. *CoRR*, abs/1807.07156, 2018.

Fu, Y., Jiang, N., and Sun, H. Binary matrix factorization and consensus algorithms. In *ICECE*, pp. 4563–4567, 2010.

Gutch, H. W., Gruber, P., Yeredor, A., and Theis, F. J. ICA over finite fields - separability and algorithms. *Signal Processing*, 92 (8):1796–1808, 2012.

Jiang, P., Peng, J., Heath, M., and Yang, R. A clustering approach to constrained binary matrix factorization. In *Data Mining and Knowledge Discovery for Big Data*, pp. 281–303. Springer, 2014.

Kane, D. M., Nelson, J., and Woodruff, D. P. An optimal algorithm for the distinct elements problem. In *PODS*, pp. 41–52, 2010.

Kanungo, T., Mount, D. M., Netanyahu, N. S., Piatko, C. D., Silverman, R., and Wu, A. Y. A local search approximation algorithm for k-means clustering. *Computational Geometry*, 28 (2-3):89–112, 2004.

Koyutürk, M. and Grama, A. PROXIMUS: a framework for analyzing very high dimensional discrete-attributed datasets. In *KDD*, pp. 147–156, 2003.

Lu, H., Vaidya, J., Atluri, V., and Hong, Y. Constraint-aware role mining via extended Boolean matrix decomposition. *IEEE Trans. Dependable Sec. Comput.*, 9(5):655–669, 2012.

Miettinen, P., Mielikäinen, T., Gionis, A., Das, G., and Mannila, H. The discrete basis problem. In *PKDD*, pp. 335–346, 2006.

Mitra, B., Sural, S., Vaidya, J., and Atluri, V. A survey of role mining. *ACM Comput. Surv.*, 48(4):50:1–50:37, 2016.

Neumann, S. Bipartite stochastic block models with tiny clusters. In *NeurIPS*, pp. 3871–3881, 2018.

Orlin, J. Contentment in graph theory: covering graphs with cliques. In *Indagationes Mathematicae (Proceedings)*, volume 80(5), pp. 406–424. Elsevier, 1977.

Painsky, A., Rosset, S., and Feder, M. Generalized independent components analysis over finite alphabets. *IEEE TOIT*, 62(2): 1038–1053, 2015.

Razenshteyn, I., Song, Z., and Woodruff, D. P. Weighted low rank approximations with provable guarantees. In *STOC*, pp. 250–263, 2016.

Seppänen, J. K., Bingham, E., and Mannila, H. A simple algorithm for topic identification in 0–1 data. In *PKDD*, pp. 423–434, 2003.

Shen, B., Ji, S., and Ye, J. Mining discrete patterns via binary matrix factorization. In *KDD*, pp. 757–766, 2009.

Šingliar, T. and Hauskrecht, M. Noisy-or component analysis and its application to link analysis. *JMLR*, 7:2189–2213, 2006.

Song, Z., Woodruff, D. P., and Zhong, P. Low rank approximation with entrywise ℓ_1-norm error. In *STOC*, pp. 688–701, 2017.

Song, Z., Woodruff, D. P., and Zhong, P. Towards a zero-one law for entrywise low rank approximation. *CoRR*, abs/1811.01442, 2018.

Vaidya, J., Atluri, V., and Guo, Q. The role mining problem: finding a minimal descriptive set of roles. In *SACMAT*, pp. 175–184, 2007.

Woodruff, D. P. Sketching as a Tool for Numerical Linear Algebra. *FnT-TCS*, 10(1-2):1–157, 2014.

Yeredor, A. Independent component analysis over Galois fields of prime order. *IEEE TOIT*, 57(8):5342–5359, 2011.

Zhang, Z., Li, T., Ding, C. H. Q., and Zhang, X. Binary matrix factorization with applications. In *ICDM*, pp. 391–400, 2007.