Faster Algorithms for Binary Matrix Factorization (Supplementary Material)

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

1. Proof of Theorem 7

Remark 1 Theorem 6 follows from the extension of the algorithm for $p = 1$ described in the Introduction of Charikar et al. (2002) to the setting with a relaxed triangle inequality, and using the relaxed triangle inequality $\|x-z\|_p^p \leq 2^{p-1}(\|x-y\|_p^p + \|y-z\|_p^p)$ for pth powers of p-norms for any points x, y, and z.

Suppose we are given an instance $A \in \{0,1\}^{m\times n}$ of the Bipartite Clique Partition problem with parameter k, and a real number $p \geq 1$. We first run the algorithm of Theorem 6 on the rows of A with parameter 2^k. That is, we treat the m rows of A as our pointset P of m points in $\mathbb{R}^n$. By the guarantee of Theorem 6, we output $(C_1, \ldots, C_{2^k})$ and $(c_1, \ldots, c_{2^k})$ for which

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

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 $\|x - c_i\|_p$ is minimized.

We have,

$$\begin{aligned}
&\sum_{i=1}^{2^k} \sum_{x\in C_i} \|x - d_i\|_p^p \\
&\quad\leq\quad 2^{p-1} \sum_{i=1}^{2^k} \sum_{x\in C_i} (\|x - c_i\|_p^p + \|c_i - d_i\|_p^p) \\
&\quad\leq\quad 2^p \sum_{i=2}^{2^k} \sum_{x\in C_i} \|x - c_i\|_p^p \quad\leq\quad 2^p \kappa_p \mathrm{OPT}_{2^k},
\end{aligned}$$

where the first inequality is the approximate triangle inequality for pth powers, the second inequality uses our choice of d_i, and the final inequality uses our guarantee on the c_i.

We can define a row cluster indicator matrix $I(A)$ as follows: for each row of A, if the row is in C_i, then we replace it with the point d_i.

Notice that

$$\|I(A) - A\|_p^p \leq \kappa_p \mathrm{OPT}_{2^k}, \tag{1}$$

and further that $I(A)$ is binary and has at most 2^k distinct rows. Further, note that 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,

$$\|U \cdot V - A\|_p^p \leq \mathrm{OPT}_{2^k}. \tag{2}$$

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

$$\|U \cdot V - I(A)\|_p^p \leq C \cdot \min_{U',V'} \|U' \cdot V' - I(A)\|_p^p, \tag{3}$$

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:

$$\begin{aligned}
\|U \cdot V - A\|_p^p \;&\leq\; 2^{p-1}(\|U \cdot V - I(A)\|_p^p + \|I(A) - A\|_p^p) \\
&\leq\; 2^{p-1}(C \min_{U',V'} \|U' \cdot V' - I(A)\|_p^p + \|I(A) - A\|_p^p) \\
&\leq\; 2^{p-1}(C \min_{U',V'} 2^{p-1}(\|U' \cdot V' - A\|_p^p + \|A - I(A)\|_p^p)) \\
&\quad\; + 2^{p-1}\|I(A) - A\|_p^p \\
&\leq\; 2^{2p-2} C \min_{U',V'} \|U' \cdot V' - A\|_p^p \\
&\quad\; + (2^{2p-2}C + 2^{p-1})\|A - I(A)\|_p^p \\
&\leq\; 2^{2p-2} C \min_{U',V'} \|U' \cdot V' - A\|_p^p \\
&\quad\; + (2^{2p-2}C + 2^{p-1})\kappa_p \mathrm{OPT}_{2^k} \\
&\leq\; (2^{2p-2}C + (2^{2p-2}C + 2^{p-1})\kappa_p) \min_{U',V'} \|U' \cdot V' - A\|_1,
\end{aligned}$$

where the first inequality is the approximate triangle inequality, the second inequality uses (3), the third inequality is the approximate triangle inequality, the fourth inequality rearranges terms, the fifth inequality uses (1), and the last inequality uses (2). Hence, for constant C and p, this particular U and V would provide a constant factor approximation to the Bipartite Clique Partition problem.

[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 36th International Conference on Machine Learning, Long Beach, California, PMLR 97, 2019.

Remark 2 It is tempting to try to force $I(A)$ to have a small number of distinct columns in addition to a small number of distinct rows, but this in general may not be possible. Indeed, after finding $I(A)$ with at most 2^k distinct rows, if one then runs the algorithm of Theorem 6 on $I(A)$ to obtain another matrix with at most 2^k distinct columns, then the number of distinct rows could be much larger than 2^k.

It remains to find U and V satisfying (3) for a constant $C \geq 1$. To do so, we will use recent advances in linear algebra, in particular the following theorem concerning so-called Lewis weight sampling.

Theorem 3 (Theorem 7.1 of Cohen & Peng (2015)) *For any $m \times k$ matrix U and $m \times n$ matrix B, there exists a subset S of $r = O(k^{\lceil p/2 \rceil} \log k)$ rows and $r \times r$ diagonal matrix D with entries between 1 and $\mathrm{poly}(m)$ for which*

1. *Simultaneously for all vectors $x \in \mathbb{R}^k$,*
$$\|DU_S x\|_p^p = (1 \pm 1/2)\|Ux\|_p^p,$$
2. $\|DB_S\|_p^p \leq 4\|B\|_p^p$,
3. *and the entries of D are powers of 2 and between 1 and $\mathrm{poly}(m)$.*

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

Remark 4 Although Lewis weight sampling in Theorem 3 is not usually stated in this form, it can be deduced from standard properties of Lewis weights. Namely, it is known that if one samples $O(k^{\lceil p/2 \rceil} \log k)$ rows of U according to the Lewis weights of U, and forms a sampling and rescaling matrix D, where the jth row of D is equal to $1/p_i$ 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 over-estimates to the actual values, and this rounding just increases the size of S by a factor of 2. Also, clearly all entries of D are at least 1. Also, with high probability we do not choose any i for which $p_i \leq 1/\Theta(mk \log k)$, implying the third property. For the second property, it suffices to observe that for any matrix B, $E[\|DB_S\|_p^p] = \|B\|_p^p$, and to apply a Markov bound.

The algorithm first "guesses" S and D. There are at most 2^k distinct rows of $I(A)$. The number of distinct subsets of $O(k^{\lceil p/2 \rceil} \log k)$ rows is at most $2^{O(k^{\lceil p/2 \rceil + 1} \log k)}$, so we try all such subsets S of rows. We also guess all possibilities of the corresponding D, and by Remark 4, there are only $O(\log m)^{O(k^{\lceil p/2 \rceil} \log k)}$ total guesses, which is $2^{O(k^{\lceil p/2 \rceil} \log k \log\log m)}$. If $\log m \leq 2^k$, this is still $2^{O(k^{\lceil p/2 \rceil + 1} \log k)}$ time; otherwise $\log m > 2^k$, and so $O(\log m)^{O(k^{\lceil p/2 \rceil}) \log k} = 2^{O(\log^2 \log m) \log\log\log m} \leq m$. Consequently, the time is always at most $2^{O(k^{\lceil p/2 \rceil + 1} \log k)} \mathrm{poly}(mn)$.

For each guess of S and D, we next guess DU_S^*. Since U^* is a binary matrix, and we know D, this is just $2^{O(k^{\lceil p/2 \rceil + 1} \log k)}$ guesses. We know $DI(A)_S$, and so can solve for $\mathrm{argmin}_V \|DU_S^* V - DI(A)_S\|_p^p$. 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 $\mathrm{argmin}_U \|UV - I(A)\|_p^p$, 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 and V which minimize $\|UV - I(A)\|_p^p$ over all possible guesses that we find.

Consider the right guess, so that $\|DU_S^* y\|_p^p = (1 \pm 1/2)\|U^* y\|_p^p$ for all vectors y, and also, by Theorem 3, we can assume that for the matrix $B = U^*V^* - I(A)$, we have $\|DB_S\|_p^p \leq 2\|B\|_p^p$. 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 it is:

$$\begin{aligned}
\|UV - I(A)\|_p^p &\leq \|U^*V - I(A)\|_p^p \\
&\leq 2^{p-1}(\|U^*V - U^*V^*\|_p^p + \|U^*V^* - I(A)\|_p^p) \\
&\leq 2^{p-1}((3/2)\|DU_S^*V - DU_S^*V^*\|_p^p + \|U^*V^* - I(A)\|_p^p) \\
&\leq 2^{p-1}((3/2)2^{p-1}\|DU_S^*V - DI(A)_S\|_p^p \\
&\quad +(3/2)2^{p-1}\|DI(A)_S - DU_S^*V^*\|_p^p \\
&\quad +\|U^*V^* - I(A)\|_p^p) \\
&\leq (3/2)2^{2p-2}\|DU_S^*V^* - DI(A)_S\|_p^p \\
&\quad +(3/2)2^{2p-2}\|DI(A)_S - DU_S^*V^*\|_p^p \\
&\quad +2^{p-1}\|U^*V^* - I(A)\|_p^p \\
&\leq 4 \cdot 2 \cdot (3/2)2^{2p-2}\|U^*V^* - I(A)\|_p^p \\
&\quad +2^{p-1}\|U^*V^* - I(A)\|_p^p \\
&= (122^{2p-2} + 2^{p-1})\|U^*V^* - I(A)\|_p^p,
\end{aligned}$$

where the first inequality follows since our choice of U was optimal for the given V that we found, the second inequality is the approximate triangle inequality, the third inequality follows from Property 1 of Theorem 3, the fourth inequality is the approximate triangle inequality, the fifth inequality follows from our choice of V which was optimal with respect to DU_S^* and $DI(A)_S$, and the sixth inequality follows from Property 2 of Theorem 3.

Thus, we have found U and V satisfying (3) for a constant $C \geq 1$ (depending on p), which completes the proof.

2. Proof of Theorem 9

We can now describe our algorithm. First note that $I(A)$ only has 2^k distinct rows. We partition the rows of $I(A)$ into $r = O(\log m)$ groups $G^1, G^2, \ldots, G^r$, where G^i consists of the subset of distinct rows of $I(A)$ which have a number of occurrences in the range $[2^{i-1}, 2^i)$ in $I(A)$. Consider:

$$\min_{U^1,\ldots,U^r \in \{0,1\}^{m\times k}, V^1,\ldots,V^r \in \{0,1\}^{k\times n}} \sum_{i=1}^{r} 2^i \|U^i V^i - G^i\|_F^2. \quad (4)$$

If we solve (4), then we can define V to be the concatenation of rows of $V^1, \ldots, V^r$, and then each row of U will consist of 0s together with a row of U^i, in the appropriate place for exactly one U^i, depending on which group G^i the current row of $I(A)$ we are trying to fit is in. Thus, by solving (4), we obtain a rank $O(k \log m)$ bicriteria solution $U \in \{0,1\}^{m\times O(k\log m)}$ and $V \in \{0,1\}^{O(k\log m)\times n}$ to our original problem, which will be an overall constant factor approximation. Note also to solve (4), we can solve each problem $\min_{U^i \in \{0,1\}^{m\times k}, V^i \in \{0,1\}^{k\times n}} \|U^i V^i - G^i\|_F^2$ independently, for each $i = 1, \ldots, r$. If we solve a single such problem with probability $9/10$, we can repeat it $O(\log(nm))$ times and choose the best solution found to solve each such problem with probability $1 - 1/(mn)$, at which point we can assume we solve all problems simultaneously, by a union bound.

To solve the ith such problem, crucially G^i has at most 2^k distinct rows. Therefore, we can apply Theorem 8 with the $\log_2 m$ of that theorem equal to k. Our algorithm thus samples $S^0, S^1, \ldots, S^k$ from the distribution given in the proof of Theorem 8, and with probability at least $9/10$, each of the two properties of Theorem 8 hold, where here we choose X_0 to $(V^*)^i$, the optimal solution to the ith problem, and choose B to be $(U^*)^i(V^*)^i - A$, where $(U^*)^i(V^i)^i$ is the optimal solution to the i-th problem.

The algorithm does not know $(U^*)^i$ so it guesses $S^j(U^*)^i$ for $j = 0, 1, \ldots, k$. Note that for each j, this is a binary $O(k) \times k$ matrix (recall that multiplication is over GF(2)), and so there are $2^{O(k^2)}$ guesses per j, and $2^{O(k^3)}$ guesses in total across all j. Let $S(U^*)^i$ be the matrix obtained by stacking the rows of $S^j(U^*)^i$ on top of each other, for $j = 0, 1, \ldots, k$. Let D be the fixed diagonal matrix with diagonal entries $\left(\frac{2^i}{k}\right)^{1/2}$ on the ith block, so that by the guarantees of Theorem 8, for our correct guess (and with probability at least $9/10$ over the choice of $S^0, \ldots, S^k$):

1. Simultaneously for all vectors $x \in \{0,1\}^k$, $\|D[S(U^*)^i x]\|_2^2 \geq \frac{1}{200} \cdot \|U^* x\|_2^2$, and
2. $\|D[S(U^*)^i(V^*)^i] - D[SA]\|_F^2 \leq 100\|(U^*)^i(V^*)^i - A\|_F^2$,

where S is the matrix obtained by stating the rows of S^j on top of each other. It is important to note that the multiplication by D is done *over the reals*, which is why we have used the $[\cdot]$ notation, though other multiplications are done over GF(2).

In the algorithm, given $S(U^*)^i$, we solve $\min_{V^i} \|D[S(U^*)^i V^i - SA]\|_F^2$ by solving for each column of V^i one at a time. Each column is found by trying all 2^k possibilities, giving $2^k \mathrm{poly}(mn)$ time in total to find V^i. Given V^i, we then solve $\min_{U^i} \|U^i V^i - A\|_F^2$ by solving for each row of U^i one at a time. In total this takes $2^k \mathrm{poly}(mn)$ time.

The cost of the solution we find is upper bounded by the cost for the right guess of $S(U^*)^i$, which is:

$$\begin{aligned}
\|U^i V^i - I(A)\|_F &\leq \|(U^*)^i V^i - I(A)\|_F \\
&\leq \|(U^*)^i V^i - (U^*)^i (V^*)^i\|_F + \|(U^*)^i (V^*)^i - I(A)\|_F \\
&\leq 200\|D[S(U^*)^i (V^i - (V^*)^i)\|_F + \|(U^*)^i (V^*)^i - I(A)\|_F \\
&\leq 200\|D[S(U^*)^i V^i - D[SI(A)]\|_F \\
&\quad +200\|D[SI(A)] - D[S(U^*)^i (V^*)^i]\|_F \\
&\quad + \|(U^*)^i (V^*)^i - I(A)\|_F \\
&\leq 200\|D[S(U^*)^i (V^*)^i - D[SI(A)]\|_F \\
&\quad +200\|D[SI(A)] - D[S(U^*)^i (V^*)^i]\|_F \\
&\quad + \|(U^*)^i (V^*)^i - I(A)\|_F \\
&= 400\|D[S(U^*)^i (V^*)^i - D[SI(A)]\|_F \\
&\quad +\|(U^*)^i (V^*)^i - I(A)\|_F \\
&\leq 40000\|(U^*)^i (V^*)^i - I(A)\|_F \\
&\quad +\|(U^*)^i (V^*)^i - I(A)\|_F \\
&= 40001\|(U^*)^i (V^*)^i - I(A)\|_F,
\end{aligned}$$

where the first inequality follows since our choice of U^i was optimal for the given V^i that we found, the second inequality is the triangle inequality, the third inequality follows from Property 1 of Theorem 8, the fourth inequality is the triangle inequality, the fifth inequality follows from our choice of V^i which was optimal with respect to $D[S(U^*)^i]$ and $D[SI(A)]$, the first equality combines terms, the last inequality follows from Property 2 of Theorem 8, and the final equality combines terms.

Combining the above for each i, we have found U and V satisfying (3) for a constant $C \geq 1$, which completes the proof.

References

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.

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