Jointly Clustering Rows and Columns of Binary Matrices: Algorithms and Trade-offs

Jiaming Xu
Electrical and Computer Engineering
University of Illinois at Urbana-Champaign
Urbana,IL,61801
jxu18@illinois.edu

Rui Wu
Electrical and Computer Engineering
University of Illinois at Urbana-Champaign
Urbana,IL,61801
ruiwu1@illinois.edu

Kai Zhu
School of Electrical, Computer and Energy Engineering
Arizona State University
Tempe, AZ 85287
kzhu17@asu.edu

Bruce Hajek
Electrical and Computer Engineering
University of Illinois at Urbana-Champaign
Urbana,IL,61801
b-hajek@illinois.edu

R. Srikant
Electrical and Computer Engineering
University of Illinois at Urbana-Champaign
Urbana,IL,61801
rsrikant@illinois.edu

Lei Ying
School of Electrical, Computer and Energy Engineering
Arizona State University
Tempe, AZ 85287
lei.ying.2@asu.edu

ABSTRACT

In standard clustering problems, data points are represented by vectors, and by stacking them together, one forms a data matrix with row or column cluster structure. In this paper, we consider a class of binary matrices, arising in many applications, which exhibit both row and column cluster structure, and our goal is to exactly recover the underlying row and column clusters by observing only a small fraction of noisy entries. We first derive a lower bound on the minimum number of observations needed for exact cluster recovery. Then, we study three algorithms with different running time and compare the number of observations needed by them for successful cluster recovery. Our analytical results show smooth time-data trade-offs: one can gradually reduce the computational complexity when increasingly more observations are available.

Categories and Subject Descriptors

I.5.3 [**PATTERN RECOGNITION**]: Clustering—*Algorithms*; G.1.6 [**NUMERICAL ANALYSIS**]: Optimization—*Convex programming*; G.3 [**PROBABILITY AND STATISTICS**]: [Statistical computing]

General Terms

Theory, Algorithms, Performance

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

SIGMETRICS'14, June 16–20, 2014, Austin, Texas, USA.
Copyright is held by the owner/author(s). Publication rights licensed to ACM.
ACM 978-1-4503-2789-3/14/06 ...$15.00.
http://dx.doi.org/10.1145/2591971.2592005.

Keywords

Clustering; Low-Rank Matrix Recovery; Spectral Method

1. INTRODUCTION

Data matrices exhibiting both row and column cluster structure, arise in many applications, such as collaborative filtering, gene expression analysis, and text mining. For example, in recommender systems, a rating matrix can be formed with rows corresponding to users and columns corresponding to items, and similar users and items form clusters. In DNA microarrays, a gene expression matrix can be formed with rows corresponding to patients and columns corresponding to genes, and similar patients and genes form clusters. Such row and column cluster structure is of great scientific interest and practical importance. For instance, the user and movie cluster structure is crucial for predicting user preferences and making accurate item recommendations [31]. The patient and gene cluster structure reveals functional relations among genes and helps disease detection [33, 26]. In practice, we usually only observe a very small fraction of entries in these data matrices, possibly contaminated with noise, which obscures the intrinsic cluster structure. For example, in Netflix movie dataset, about 99% of movie ratings are missing and the observed ratings are noisy [29].

In this paper, we study the problem of inferring hidden row and column cluster structure in binary data matrices from a few noisy observations. We consider a simple model introduced in [1, 4] for generating binary data matrix from underlying row and column clusters. In the context of movie recommender systems, our model assumes that users and movies each form equal-sized clusters. Users in the same cluster give the same rating to movies in the same cluster, where ratings are either $+1$ or -1 with $+1$ being "like" and -1 being "dislike". Each rating is flipped independently with a fixed flipping probability less than $1/2$, modeling the noisy user behavior and the fact that users (movies) in the same cluster do not necessarily give (receive) identical ratings. Each rating is further erased independently with a erasure probability, modeling the fact that some ratings are not observed. Then, from the observed noisy ratings, we aim to *exactly* recover the underlying user and movie clusters, i.e., jointly cluster the rows and columns of the observed rating matrix.

The binary assumption on data matrices is of practical interest. Firstly, in many real datasets like Netflix dataset and DNA microarrays, estimation of entry values appears to be very unreliable, but the task of determining whether an entry is $+1$ or -1 can be done more reliably [4]. Secondly, in recommender systems like rating music on Pandora or rating posts on sites such as Facebook and MathOverflow, the user ratings are indeed binary [21]. The equal-sized assumption on cluster size is just for ease of presentation and can be relaxed to allow for different cluster sizes.

The hardness of our cluster recovery problem is governed by the erasure probability and cluster size. Intuitively, cluster recovery becomes harder when the erasure probability increases, meaning fewer observations, and the cluster size decreases, meaning that clusters are harder to detect. The first goal of this paper is to understand when exact cluster recovery is possible or fundamentally impossible. Furthermore, our cluster recovery problem poses a computational challenge: An algorithm exhaustively searching over all the possible row and column cluster structures would have a time complexity exponentially increasing with the matrix dimension. The second goal of this paper is to understand how the computational complexity of our cluster recovery problem changes when increasingly more observations are available.

In this paper, our contributions are as follows. We first derive a lower bound on the minimum number of observations needed for exact cluster recovery as a function of matrix dimension and cluster size. Then we propose three algorithms with different running times and compare the number of observations needed by them for successful cluster recovery.

- The first algorithm directly searches for the optimal clustering of rows and columns separately; it is combinatorial in nature and takes exponential-time but achieves the best statistical performance among the three algorithms in the noiseless setting.
- By noticing that the underlying true rating matrix is a specific type of low-rank matrix, the second algorithm recovers the clusters by solving a nuclear norm regularized convex optimization problem, which is a popular heuristic for low rank matrix completion problems; it takes polynomial-time but has less powerful statistical performance than the first algorithm.
- The third algorithm applies spectral clustering to the rows and columns separately and then performs a joint clean-up step; it has lower computational complexity than the previous two algorithms, but less powerful statistical performance. We believe that this is the first such performance guarantee for exact cluster recovery, with a growing number of clusters, using spectral clustering.

These algorithms are then compared with a simple nearest-neighbor clustering algorithm proposed in [4]. Our analytical results show smooth time-data trade-offs: when increasingly more observations are available, one can gradually reduce the computational complexity by applying simpler algorithms while still achieving the desired performance. Such time-data trade-offs is of great practical interest for statistical learning problems involving large datasets [13]. Furthermore, we observed that our exponential-time combinatorial method needs substantially few observations for successful cluster recovery than the other three polynomial-time counterparts, which suggests that a large performance gap might exist between exponential-time algorithms and polynomial-time algorithms. Similar performance gap due to the computational complexity constraint has also been observed recently in many other inference problems such as graph clustering [2, 19], Sparse PCA [6, 5, 30] and sparse submatrix detection [28, 3, 32].

The rest of the paper is organized as follows. In Section 2, we discuss related work. In Section 3, we formally introduce our model and main results. The lower bound is presented in Section 4. The combinatorial method, convex method, spectral method are studied in Section 5, Section 6 and Section 7, respectively. The proofs are given in Section 8. The simulation results are presented in Section 9. Section 10 concludes the paper with remarks.

2. RELATED WORK

In this section, we point out some connections of our model and results to prior work. There is a vast literature on clustering and we only focus on theoretical works with rigorous performance analysis. More detailed comparisons are provided after we present the theorems.

2.1 Graph clustering

Much of the prior work on graph clustering, as surveyed in [22], focuses on graphs with a single node type, where nodes in the same cluster are more likely to have edges among them. A low-rank plus sparse matrix decomposition approach is proved to exactly recover the clusters with the best known performance guarantee in [18]. The same approach is used to recover the clusters from a partially observed graph in [17]. A spectral method for exact cluster recovery is proposed and analyzed in [34] with the number of clusters fixed. More recently, [36] proved an upper bound on the number of nodes "mis-clustered" by a spectral clustering algorithm in the high dimensional setting with a growing number of clusters. An interesting recent work [41] studies the graph clustering problem under both non-adaptive and adaptive sampling strategies of node pairs.

In contrast to the above works, in our model, we have a labeled bipartite graph with two types of nodes (rows and columns). Notice that there are no edges among nodes of the same type and cluster structure is defined for the two types separately. In this sense, our cluster recovery problem can be viewed as a natural generalization of graph clustering problem to labeled bipartite graphs. In fact, our second algorithm via convex programming is inspired by the work [18, 17, 19].

A model similar to ours but with a fixed number of clusters has been considered in [38], where the spectral method plus majority voting is shown to *approximately* predict the rating matrix. However, our third algorithm via spectral method is shown to achieve exact cluster and rating matrix recovery with a growing number of clusters. This is the first theoretical result on spectral method for exact cluster recovery with a growing number of clusters to our knowledge.

2.2 Biclustering

Biclustering [23, 20, 33, 8] tries to find (overlap) submatrices with particular patterns in a data matrix. Many of the proposed algorithms are based on heuristic searches without provable performance guarantees. Our cluster recovery problem can be viewed as a special case where the data matrix consists of non-overlapping sub-matrices with constant binary entries, and our paper provides a thorough study of this special biclustering problem. Recently, there is a line of work studying another special case of biclustering problem, which tries to detect a single small submatrix with elevated mean in a large fully observed noisy matrix [28]. Interesting statistical and computational trade-offs are summarized in [3].

2.3 Low-rank matrix completion

Under our model, the underlying true data matrix is a specific type of low-rank matrix. If we recover the true data matrix, we immediately get the user (or movie) clusters by assigning the identical rows (or columns) of the matrix to the same cluster. In the noiseless setting with no flipping, the nuclear norm minimization approach [12, 11, 35] can be directly applied to recover the true data matrix and further recover the row and column clusters. Alternate minimization is another popular and empirically successful approach for low-matrix completion [29]. However, it is harder to analyze and the performance guarantee is weaker than nuclear norm minimization [24]. In the low noise setting with the flipping probability restricting to be a small constant, the low-rank plus sparse matrix decomposition approach [14, 10, 16] can be applied to exactly recover data matrix and further recover the row and column clusters.

The performance guarantee for our convex method is better than these previous approaches and it allows the flipping probability to be any constant less than 1/2. Moreover, our proof turns out to be much simpler. The recovery of our true data matrix from binary observations can also be viewed as a specific type of one-bit matrix completion problem recently studied in [21]. However, [21] focuses on approximately recovering a low rank matrix with real-valued entries.

3. MODEL AND MAIN RESULTS

In this section, we formally state our model and main results.

3.1 Model

Our model is described in the context of movie recommender systems, but it is applicable to other systems with binary data matrices having row and column cluster structure. Consider a movie recommender system with n users and n movies. Let R be the rating matrix of size $n \times n$ where R_{ij} is the rating user i gives to movie j. Assume both users and movies form r clusters of size $K = n/r$. Users in the same cluster give the same rating to movies in the same cluster. The set of ratings corresponding to a user cluster and a movie cluster is called a block. Let B be the *block rating matrix* of size $r \times r$ where B_{kl} is the *block rating* user cluster k gives to movie cluster l. Then the rating $R_{ij} = B_{kl}$ if user i is in user cluster k and movie j is in movie cluster l. Further assume that entries of B are independent random variables which are $+1$ or -1 with equal probability. Thus, we can imagine the rating matrix as a block-constant matrix with all the entries in each block being either $+1$ or -1. Observe that if r is a fixed constant, then users from two different clusters have the same ratings for all movies with some positive probability, in which case it is impossible to differentiate between these two clusters. To avoid such situations, assume r is at least $\Omega(\log n)$.

Suppose each entry of R goes through an independent binary symmetric channel with flipping probability $p < 1/2$, representing noisy user behavior, and an independent erasure channel with erasure probability ϵ, modeling the fact that some entries are not observed. The expected number of observed ratings is $m = n^2(1-\epsilon)$. We assume that p is a constant throughout the paper and ϵ could converge to 1 as $n \to \infty$. Let R' denote the output of the binary symmetric channel and Ω denote the set of non-erased entries. Let $\widehat{R}_{ij} = R'_{ij}$ if $(i,j) \in \Omega$ and $\widehat{R}_{ij} = 0$ otherwise. The goal is to exactly recover the row and column clusters from the observation $\widehat{R}$.

3.2 Main Results

The main results are summarized in Table 1. Note that these results do not explicitly depend on p. In fact, as p is assumed to be a constant strictly less than 1/2, it affects the results by constant factors.

The parameter regime where exact cluster recovery is fundamentally impossible for any algorithm is proved in Section 4. The combinatorial method, convex method and spectral method are studied in Section 5, Section 6 and Section 7, respectively. We only analyze the combinatorial method in the noiseless case where $p = 0$, but we believe similar result is true for the noisy case as well. The parameter regime in which the convex method succeeds is obtained by assuming that a technical conjecture holds, which is justified through extensive simulation. The parameter regime in which the spectral method succeeds is obtained for the first time for exact cluster recovery with a growing number of clusters. The nearest-neighbor clustering algorithm was proposed in [4]. It clusters the users by finding the $K-1$ most similar neighbors for each user. The similarity between user i and i' is measured by the number of movies with the same observed rating, i.e.,

$$s_{ii'} = \sum_{j=1}^{n} \mathbb{I}_{\{\widehat{R}_{ij} \neq 0\}} \mathbb{I}_{\{\widehat{R}_{i'j} \neq 0\}} \mathbb{I}_{\{\widehat{R}_{ij} = \widehat{R}_{i'j}\}},$$

where $\mathbb{I}_{\{\cdot\}}$ is an indicator function. Movies are clustered similarly. It is shown in [4] that the nearest-neighbor clustering algorithm exactly recovers user and movie clusters when $n(1-\epsilon)^2 > C \log n$ for a constant C.

The number of observations needed for successful cluster recovery can be derived from the corresponding parameter regime using the identity $m = n^2(1-\epsilon)$ as shown in Table 1. For better illustration, we visualize our results in Figure 1. In particular, we take $\log(m/n)$ as x-axis and $\log K$ as y-axis and normalize both axes by $\log n$. Since exact cluster recovery becomes easy when the number of observations m and cluster size K increase, we expect that exact cluster recovery is easy near $(1, 1)$ and hard near $(0, 0)$.

From Figure 1, we can observe interesting trade-offs between algorithmic running time and statistical performance. In terms of the running time, the combinatorial method is exponential, while the other three algorithms are polynomial. In particular, the convex method can be casted as a semidefinite programming and solved in polynomial-time. For the spectral method, the most computationally expen-

	regime in (K,ϵ)	regime in (K,m)	running time	remark
lower bound	$nK^2(1-\epsilon)^2 = O(1)$	$m = O(\frac{n^{1.5}}{K})$		
combinatorial method	$nK(1-\epsilon)^2 = \Omega(\log n)$	$m = \Omega(\frac{n^{1.5}\sqrt{\log n}}{\sqrt{K}})$	exponential	assuming noiseless
convex method	$K(1-\epsilon) = \Omega(\log n)$	$m = \Omega(\frac{n^2 \log n}{K})$	polynomial	assuming Conjecture 1
spectral method	$K^2(1-\epsilon) = \Omega(n \log^2 n)$	$m = \Omega(\frac{n^3 \log^2 n}{K^2})$	$O(n^3)$	
nearest-neighbor clustering	$n(1-\epsilon)^2 = \Omega(\log n)$	$m = \Omega(n^{1.5}\sqrt{\log n})$	$O(mr)$	

Table 1: Main results: comparison of a lower bound and four clustering algorithms.

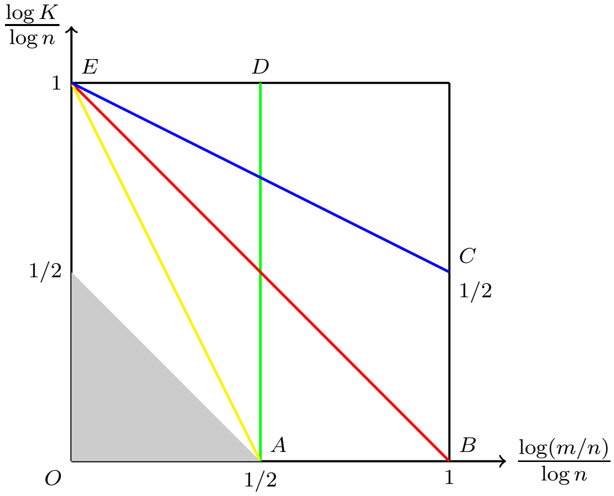

Figure 1: Summary of results in terms of number of observations m and cluster size K. The lower bound states that it is impossible for any algorithm to reliably recover the clusters exactly in the shaded regime (grey). The combinatorial method, the convex method, the spectral method and the nearest-neighbor clustering algorithm succeed in the regime to the right of lines AE (yellow), BE (red), CE (blue) and AD (green), respectively.

sive step is the singular value decomposition of the observed data matrix which can always be done in time $O(n^3)$ and more efficiently when the observed data matrix is sparse. It is not hard to see that the time complexity for the nearest-neighbor clustering algorithm is $O(n^2 r)$ and more careful analysis reveals that its time complexity is $O(mr)$. On the other hand, in terms of statistical performance, the combinatorial method needs strictly fewer observations than the other three algorithms when there is no noise, and the convex method always needs fewer observations than the spectral method. It is somewhat surprising to see that the simple nearest-neighbor clustering algorithm needs fewer observations than the more sophisticated convex method when the cluster size K is $O(\sqrt{n})$.

In summary, we see that when more observations available, one can apply algorithms with less running time while still achieving exact cluster recovery. For example, consider the noiseless case with cluster size $K = n^{0.8}$, the number of observations *per user* required for cluster recovery by the combinatorial method, convex method, spectral method and nearest-neighbor clustering algorithm are $\Omega(n^{0.1})$, $\Omega(n^{0.2})$, $\Omega(n^{0.4})$ and $\Omega(n^{0.5})$, respectively. Therefore, when the number of observations per user increases from $\Omega(n^{0.1})$ to $\Omega(n^{0.5})$, one can gradually reduces the computational complexity from exponential-time to polynomial-time as low as $O(n^{1.7})$.

The main results in this paper can be easily extended to the more general case with n_1 rows and $n_2 = \Theta(n_1)$ columns and r_1 row clusters and $r_2 = \Theta(r_1)$ column clusters. The sizes of different clusters could vary as long as they are of the same order. Likewise, the flipping probability p and the erasure probability ϵ could also vary for different entries of the data matrix as long as they are of the same order. Due to space constraints, such generalizations are omitted in this paper.

3.3 Notations

A variety of norms on matrices will be used. The spectral norm of a matrix X is denoted by $\|X\|$, which is equal to the largest singular value. Let $\langle X, Y\rangle = \mathrm{Tr}(X^\top Y)$ denote the inner product between two matrices. The nuclear norm is denoted by $\|X\|_*$ which is equal to the sum of singular values and is a convex function of X. Let $\|X\|_1 = \sum_{i,j} |X_{ij}|$ denote the l_1 norm and $\|X\|_\infty = \max_{i,j} |X_{ij}|$ denote the l_∞ norm. Let $X = \sum_{t=1}^n \sigma_t u_t v_t^\top$ denotes the singular value decomposition $X \in \mathbb{R}^{n\times n}$ such that $\sigma_1 \geq \cdots \geq \sigma_n$. The best rank r approximation of X is defined as $P_r(X) = \sum_{t=1}^r \sigma_t u_t v_t^\top$. For vectors, let $\langle x, y\rangle$ denote the inner product between two vectors and the only norm that will be used is the usual l_2 norm, denoted as $\|x\|_2$.

Throughout the paper, we say that an event occurs "a.a.s." or "asymptotically almost surely" when it occurs with a probability which tends to one as n goes to infinity.

4. LOWER BOUND

In this section, we derive a lower bound for any algorithm to reliably recover the user and movie clusters. The lower bound is constructed by considering a genie-aided scenario where the set of flipped entries is revealed as side information, which is equivalent to saying that we are in the noiseless setting with $p = 0$. Hence, the true rating matrix R agrees with $\widehat{R}$ on all non-erased entries. We construct another rating matrix $\tilde{R}$ with the same movie cluster structure as R but different user cluster structure by swapping two users in two different user clusters. We show that if $nK^2(1-\epsilon)^2 = O(1)$, then $\tilde{R}$ agrees with $\widehat{R}$ on all non-erased entries with positive probability, which implies that no algorithm can reliably distinguish between R and $\widehat{R}$ and thus recover user clusters.

Theorem 1. *Fix $0 < \delta < 1$. If $nK^2(1-\epsilon)^2 < \delta$, then with probability at least $1-\delta$, it is impossible for any algorithms to recover the user clusters or movie clusters.*

Intuitively, Theorem 1 says that when the erasure probability is high and the cluster size is small that $nK^2(1-\epsilon)^2 = O(1)$, the observed rating matrix $\widehat{R}$ does not carry enough information to distinguish between different possible cluster structures.

5. COMBINATORIAL METHOD

In this section, we study a combinatorial method which clusters users or movies by searching for a partition with the least total number of "disagreements". We describe the method in Algorithm 1 for clustering users only. Movies are clustered similarly. The number of disagreements $D_{ii'}$ between a pair of users i, i' is defined as the number of movies satisfying that: The two ratings given by users i, i' are both observed and the observed two ratings are different. In particular, if for every movie, the two ratings given by users i, i' are not observed simultaneously, then $D_{ii'} = 0$.

Algorithm 1 Combinatorial Method

```
For each pair of users $i, i'$, compute the number of disagreements $D_{ii'}$ between them.
For each partition of users into $r$ clusters of equal size $K$, compute its total number of disagreements defined as
    $$\sum_{i,i' \text{ in the same cluster}} D_{ii'}$$
Output a partition which has the least total number of disagreements.
```

The idea of Algorithm 1 is to reduce the problem of clustering both users and movies to a standard user clustering problem without movie cluster structure. In fact, this algorithm looks for the optimal partition of the users which has the minimum total in-cluster distance, where the distance between two users is measured by the number of disagreements between them. The following theorem shows that such simple reduction does *not* achieve the lower bound given in Theorem 1. The optimal algorithm for our cluster recovery problem might need to explicitly make use of both user and movie cluster structures.

Theorem 2. *If $nK(1-\epsilon)^2 \leq \frac{1}{4}$, then with probability at least $3/4$, Algorithm 1 cannot recover user and movie clusters.*

Next we show that the above necessary condition for the combinatorial method is also sufficient up to a logarithmic factor when there is no noise, i.e., $p = 0$. We suspect that the theorem holds for the noisy setting as well, but we have not yet been able to prove this.

Theorem 3. *If $p = 0$ and $nK(1-\epsilon)^2 > C \log n$ for some constant C, then a.a.s. Algorithm 1 exactly recovers user and movie clusters.*

This theorem is proved by considering a conceptually simpler greedy algorithm that does not need to know K. After computing the number of disagreements for every pair of users, we search for a largest set of users which have no disagreement between each other, and assign them to a new cluster. We then remove these users and repeat the searching process until there is no user left. In the noiseless setting, the K users from the same true cluster have no disagreement between each other. Therefore, it is sufficient to show that, for any set of K users consisting of users from more than one cluster, they have more than one disagreement with high probability under our assumption.

6. CONVEX METHOD

In this section, we show that the rating matrix R can be exactly recovered by a convex program, which is a relaxation of the maximum likelihood (ML) estimation. When R is known, we immediately get the user (or movie) clusters by assigning the identical rows (or columns) of R to the same cluster.

Let $\mathcal{Y}$ denote the set of binary block-constant rating matrix with r^2 blocks of equal size. As the flipping probability $p < 1/2$, Maximum Likelihood (ML) estimation of R is equivalent to finding a $Y \in \mathcal{Y}$ which best matches the observation $\widehat{R}$:

$$\begin{aligned} \max_{Y} \; & \sum_{i,j} \widehat{R}_{ij} Y_{ij} \\ \text{s.t.} \; & Y \in \mathcal{Y}. \end{aligned} \tag{1}$$

Since $|\mathcal{Y}| = \Omega(e^n)$, solving (1) via exhaustive search takes exponential-time. Observe that $Y \in \mathcal{Y}$ implies that Y is of rank at most r. Therefore, a natural relaxation of the constraint that $Y \in \mathcal{Y}$ is to replace it with a rank constraint on Y, which gives the following problem:

$$\begin{aligned} \max_{Y} \; & \sum_{i,j} \widehat{R}_{ij} Y_{ij} \\ \text{s.t.} \; & \operatorname{rank}(Y) \leq r, \; Y_{ij} \in \{1, -1\}. \end{aligned}$$

Further by relaxing the integer constraint and replacing the rank constraint with the nuclear norm regularization, which is a standard technique for low-rank matrix completion, we get the desired convex program:

$$\begin{aligned} \max_{Y} \; & \sum_{i,j} \widehat{R}_{ij} Y_{ij} - \lambda \|Y\|_* \\ \text{s.t.} \; & Y_{ij} \in [-1, 1]. \end{aligned} \tag{2}$$

The clustering algorithm based on the above convex program is given in Algorithm 2

Algorithm 2 Convex Method

```
(Rating matrix estimation) Solve for $\widehat{Y}$ the convex program (2).
(Cluster estimation) Assign identical rows (columns) of $\widehat{Y}$ to the same cluster.
```

The convex program (2) can be casted as a semidefinite program and solved in polynomial-time. Thus, Algorithm 2 takes polynomial-time. Our performance guarantee for Algorithm 2 is stated in terms of the incoherence parameter μ defined below. Since the rating matrix R has rank r, the singular vector decomposition (SVD) is $R = U \Sigma V^\top$, where $U, V \in \mathbb{R}^{n \times r}$ are matrices with orthonormal columns and $\Sigma \in \mathbb{R}^{r \times r}$ is a diagonal matrix with nonnegative entries.

Define incoherence parameter $\mu > 0$ such that $\|UV^\top\|_\infty \leq \mu\sqrt{r}/n$. A small value of μ means that the left and right singular vectors of R are unaligned with each other. Denote the SVD of the block rating matrix B by $B = U_B \Sigma_B V_B^\top$. The next lemma shows that

$$\|UV^\top\|_\infty = \|U_B V_B^\top\|_\infty / K, \tag{3}$$

and thus μ is upper bounded by $\sqrt{r}$.

Lemma 1. *$\mu \leq \sqrt{r}$.*

Recent studies [12, 11, 35] in low-rank matrix completion have demonstrated that the number of samples needed for exact low-rank matrix recovery depends on the incoherence parameter μ. Not surprisingly, the performance guarantee for Algorithm 2 given by the following theorem also depends on μ.

Theorem 4. *If $n(1-\epsilon) \geq C' \log^2 n$ for some constant C', and*

$$m > Cnr \max\{\log n, \mu^2\}, \tag{4}$$

where C is a constant and μ is the incoherence parameter for R, then a.a.s. the rating matrix R is the unique maximizer to the convex program (2) with $\lambda = 3\sqrt{(1-\epsilon)n}$.

Our proof shows that with appropriate choices of λ, the nuclear norm regularization is effective in "de-noising" and the effectiveness depends on $\|UV^\top\|_\infty$. This is exactly why our performance guarantee depends on the incoherence parameter μ. Note that Algorithm 2 is easy to implement as λ only depends on the erasure probability ϵ, which can be reliably estimated from $\widehat{R}$. Moreover, the particular choice of λ in the theorem is just to simplify notations. It is straightforward to generalize our proof to show that the above theorem holds with $\lambda = C_1\sqrt{(1-\epsilon)n}$ for any constant $C_1 \geq 3$.

Using Lemma 1, we immediately conclude from the above theorem that the convex program succeeds when $m > Cnr^2$ for some constant C. However, based on extensive simulation in Fig 2, we conjecture that the following result is true.

Conjecture 1. *$\mu = \Theta(\sqrt{\log r})$ a.a.s.*

Conjecture 1 is equivalent to $\|U_B V_B^\top\|_\infty = \Theta(\sqrt{\frac{\log r}{r}})$ due to (3). For a fixed r, we simulate 1000 independent trials of B, pick the largest value of $\|U_B V_B^\top\|_\infty$, scale it by dividing $\sqrt{\log r / r}$, and get the plot in Fig 2.

Assuming this conjecture holds, Theorem 4 implies that

$$m > Cnr \log n$$

for some constant C is sufficient to recover the rating matrix, which is better than the previous condition by a factor of r. We do not have a proof for the conjecture at this time.

Comparison to previous work In the noiseless setting with $p = 0$, the nuclear norm minimization approach [12, 11, 35] can be directly applied to recover data matrix and further recover the row and column clusters. It is shown in [35] that the nuclear norm minimization approach exactly recovers the matrix with high probability if $m = \Omega(\mu^2 nr \log^2 n)$. The performance guarantee for Algorithm 2 given in (4) is better by at least a factor of $\log n$. Theorem 3 shows that the combinatorial method exactly recovers the row and column clusters if $m = \Omega(nr^{1/2} \log^{1/2} n)$, which is substantially better than the two previous conditions by at least a factor of $r^{1/2}$. This suggests that a large performance gap might exist between exponential-time algorithms and polynomial-time algorithms. Similar performance gap due to the computational complexity constraint has also been observed in other inference problems such as graph clustering [2, 19], Sparse PCA [6, 5, 30] and sparse submatrix detection [28, 3, 32].

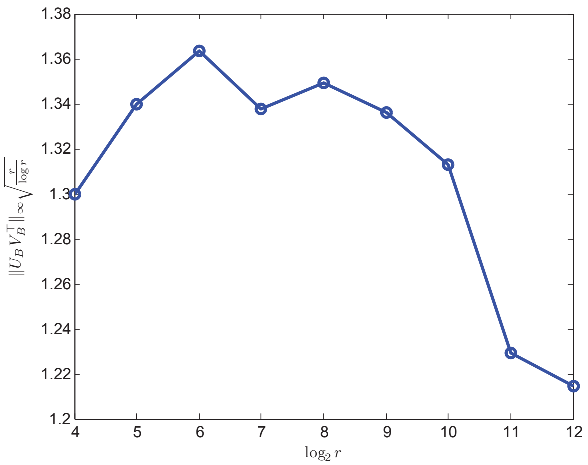

Figure 2: Simulation result supporting Conjecture 1. The conjecture is equivalent to $\|U_B V_B^\top\|_\infty = \Theta(\sqrt{\frac{\log r}{r}})$.

In the low noise setting with p restricted to be a small constant, the low-rank plus sparse matrix decomposition approach [14, 10, 16] can be applied to exactly recover data matrix and further recover the row and column clusters. It is shown in [16] that a weighted nuclear norm and l_1 norm minimization succeeds with high probability if $m = \Omega(\rho_r \mu^2 nr \log^6 n)$ and $p \leq \rho_s$ for two constants ρ_r and ρ_s. The performance guarantee for Algorithm 2 given in (4) is better by several $\log n$ factors and we allow the fraction of noisy entries p to be any constant less than $1/2$. Moreover, our proof turns out to be much simpler. The recovery of our true data matrix from binary observations can also be viewed as a specific type of one-bit matrix completion problem [21]: Given an unknown low rank-r matrix M, generate a binary matrix $Y \in \{\pm 1\}^{n \times n}$ such that $Y_{ij} = 1$ with probability $f(M_{ij})$ and the task is to recover M from a partial observation of Y. By taking $f(1) = 1-p, f(-1) = p$, our problem reduces to the one-bit matrix completion problem. It is shown in [21] that approximate recovery is possible using the maximum likelihood estimation with nuclear norm constraint. In contrast, as shown in Theorem 4, our convex method yields exact recovery.

7. SPECTRAL METHOD

In this section, we study a polynomial-time clustering algorithm based on the spectral projection of the observed rating matrix $\widehat{R}$. The description is given in Algorithm 3.

Step 1 of the algorithm produces two subsets, Ω_1 and Ω_2, of Ω such that: 1) for $i \in \{1, 2\}$, each rating is observed in Ω_i with probability $\frac{1-\epsilon}{2}$, independently of other elements; and 2) Ω_1 is independent of Ω_2. The purpose of Step 1 is to remove dependency between Step 2 and Steps 3, 4 in our

Algorithm 3 Spectral Method

```
1: (Producing two subsets, $\Omega_1$ and $\Omega_2$, of $\Omega$ via randomly sub-sampling $\Omega$) Let $\delta = \frac{1-\epsilon}{4}$, and independently assign each element of $\Omega$ only to $\Omega_1$ with probability $\frac{1}{2}-\delta$, only to $\Omega_2$ with probability $\frac{1}{2}-\delta$, to both $\Omega_1$ and $\Omega_2$ with probability $\delta$, and to neither $\Omega_1$ nor $\Omega_2$ with probability $\delta$. Let $\widehat{R}^{(1)}_{i,j} = \widehat{R}_{i,j}\mathbb{I}_{\{(i,j)\in\Omega_1\}}$ and $\widehat{R}^{(2)}_{i,j} = \widehat{R}_{i,j}\mathbb{I}_{\{(i,j)\in\Omega_2\}}$ for $i,j \in \{1,\dots,n\}$.
2: (Approximate clustering) Let $P_r(\widehat{R}^{(1)})$ denote the rank $r$ approximation of $\widehat{R}^{(1)}$ and let $x_i$ denote the $i$-th row of $P_r(\widehat{R}^{(1)})$. Construct user clusters $\widehat{C}_1,\dots,\widehat{C}_r$ sequentially as follows. For $1 \le k \le r$, after $\widehat{C}_1,\dots,\widehat{C}_{k-1}$ have been selected, choose an initial user not in the first $k-1$ clusters, uniformly at random, and let $\widehat{C}_k = \{i' : ||x_i - x_{i'}|| \le \tau\}$. (The threshold $\tau$ is specified below.) Assign each remaining unclustered user to a cluster arbitrarily. Similarly, construct movie clusters $\widehat{D}_1,\dots,\widehat{D}_r$ based on the columns of $P_r(\widehat{R}^{(1)})$.
3: (Block rating estimation by majority voting) For $k,l \in \{1,\dots,r\}$, let $\widehat{V}_{kl} = \sum_{i\in\widehat{C}_k}\sum_{j\in\widehat{D}_l}\widehat{R}^{(2)}_{ij}$ be the total vote that user cluster $\widehat{C}_k$ gives to movie cluster $\widehat{D}_l$. If $\widehat{V}_{kl} \ge 0$, let $\widehat{B}_{kl} = 1$; otherwise, let $\widehat{B}_{kl} = -1$.
4: (Reclustering by assigning users and movies to nearest centers) Recluster users as follows. For $k \in \{1,\dots,r\}$, define center $\mu_k$ for user cluster $\widehat{C}_k$ as $\mu_{kj} = \widehat{B}_{kl}$ if movie $j \in \widehat{D}_l$ for all $j$. Assign user $i$ to cluster $k$ if $\langle \widehat{R}^{(2)}_{i,\cdot}, \mu_k\rangle \ge \langle \widehat{R}^{(2)}_{i,\cdot}, \mu_{k'}\rangle$ for all $k' \ne k$. Recluster movies similarly.
```

proof. In particular, to establish our theoretical results, we identify the initial clustering of users and movies using Ω_1, and then majority voting and reclustering are done using Ω_2. In practice, one can simply use the same set of observations, i.e., $\Omega_1 = \Omega_2 = \Omega$.

The following theorem shows that the spectral method exactly recovers the user and movie clusters under a condition stronger than (4). In particular, we show that Step 3 exactly recovers the block rating matrix B and Step 4 cleans up clustering errors made in Step 2.

Theorem 5. *If*

$$n(1-\epsilon) > Cr^2\log^2 n, \tag{5}$$

for a positive constant C, then Algorithm 3 with $\tau = 12(1-\epsilon)^{1/2}r\log n$ a.a.s. exactly recovers user and movie clusters, and the rating matrix R.

Algorithm 3 is also easy to implement as τ only depends on parameters ϵ and r. As mentioned before, the erasure probability ϵ can be reliably estimated from $\widehat{R}$ using empirical statistics. The number of clusters r can be reliably estimated by searching for the largest eigen-gap in the spectrum of $\widehat{R}$ (See Algorithm 2 and Theorem 3 in [18] for justification). We further note that the threshold τ used in the theorem can be replaced by $C_1(1-\epsilon)^{1/2}r\log n$ for any constant $C_1 \ge 12$.

Comparison to previous work Variants of spectral method are widely used for clustering nodes in a graph. Step 2 of Algorithm 3 for approximate clustering has been previously proposed and it is analyzed in [27]. In [34], an adaptation of Step 1 is shown to exactly recover a fixed number of clusters under the planted partition model. More recently, [36] proves an upper bound on the number of nodes "misclustered" by spectral method under the stochastic block model with a growing number of clusters.

Compared to previous work, the main novelty of Algorithm 3 is Steps 1, 3, and 4 which allow for exact cluster recovery even with a growing number of clusters. To our knowledge, Theorem 5 provides the first theoretical result on spectral method for exact cluster recovery with a growing number of clusters.

8. PROOFS

8.1 Proof of Theorem 1

Without loss of generality, suppose users $1,3,\dots,2K-1$ are in cluster 1 and users $2,4,\dots,2K$ are in cluster 2. We construct a block-constant matrix with the same movie cluster structure as R but a different user cluster structure. In particular, under $\tilde{R}$, user 1 forms a new cluster with users $2i, i=2,\dots,K$ and user 2 forms a new cluster with users $2i-1, i=2,\dots,K$.

Let i-th row of $\tilde{R}$ be identical to the i-th row of R for all $i > 2K$. Consider all movies j in movie cluster l. If the ratings of user 1 to movies in movie cluster l are all erased, then let $\tilde{R}_{1j} = R_{2j}$ and $\tilde{R}_{ij} = R_{2j}$ for $i = 4,6,\dots,2K$; otherwise let $\tilde{R}_{1j} = R_{1j}$ and $\tilde{R}_{ij} = R_{1j}$ for $i = 4,6,\dots,2K$. If the ratings of user 2 to movies in movie cluster l are all erased, then let $\tilde{R}_{2j} = R_{1j}$ and $\tilde{R}_{ij} = R_{1j}$ for $i = 3,5,\dots,2K-1$; otherwise let $\tilde{R}_{2j} = R_{2j}$ and $\tilde{R}_{ij} = R_{2j}$ for $i = 3,5,\dots,2K-1$. From the above procedure, it follows that the first row of $\tilde{R}$ is identical to the $(2i)$-th row of $\tilde{R}$ for all $i = 2,\dots,K$, and the second row of $\tilde{R}$ is identical the $(2i-1)$-th row of $\tilde{R}$ for all $i = 2,\dots,K$.

We show that $\tilde{R}$ agrees with $\widehat{R}$ on all non-erased entries. We say that movie cluster l is conflicting between user 1 and user cluster 2 if (1) user cluster 1 and 2 have different block rating on movie cluster l; and (2) the ratings of user 1 to movies in movie cluster l are not all erased; and (3) the block corresponding to user cluster 2 and movie cluster l is not totally erased. Therefore, the probability that movie cluster l is conflicting between user 1 and user cluster 2 equals to $\frac{1}{2}(1-\epsilon^{K^2})(1-\epsilon^K)$. By the union bound,

$$\mathbb{P}\{\exists\text{conflicting movie cluster between user 1 and cluster 2}\}$$
$$\le \frac{r}{2}(1-\epsilon^{K^2})(1-\epsilon^K) \le \frac{r}{2}K^3(1-\epsilon)^2 \le \delta/2,$$

where the third inequality follows because $(1-x)^a \ge 1-ax$ for $a \ge 1$ and $x \ge 0$ and the last inequality follows from the assumption. Similarly, the probability that there exists a conflicting movie cluster between user 2 and cluster 1 is also upper bounded by $\delta/2$. Hence, with probability at least $1-\delta$, there is no conflicting movie cluster between user 1 and cluster 2 as well as between user 2 and cluster 1, and thus $\tilde{R}$ agrees with $\widehat{R}$ on all non-erased entries.

8.2 Proof of Theorem 2

Consider a genie-aided scenario where the set of flipped entries is revealed as side information, which is equivalent to saying that we are in the noiseless setting with $p = 0$. Then the true partition corresponding to the true user cluster structure has zero disagreement. Suppose that users $1,3,\dots,2K-1$ are in true cluster 1 and users $2,4,\dots,2K$

are in true cluster 2. We construct a new partition different from the true partition by swapping user 1 and user 2. In particular, under the new partition, user 1 forms a new cluster $\widehat{C}_2$ with users $2i, i = 2, \dots, K$, user 2 forms a new cluster $\widehat{C}_1$ with users $2i-1, i = 2, \dots, K$. It suffices to show that for $k = 1, 2$, any two users in $\widehat{C}_k$ has zero disagreement with probability at least 3/4, in which case the new partition has zero agreement and Algorithm 1 cannot distinguish between the true partition and the new one.

For $k = 1, 2$, we lower bound the probability that any two users in $\widehat{C}_k$ has zero disagreement.

$$\begin{aligned}
&\mathbb{P}(\text{Any two users in } \widehat{C}_k \text{ has zero disagreement})\\
=&1 - \mathbb{P}(\text{total number of disagreements in } \widehat{C}_k \geq 1)\\
\geq&1 - \mathbb{E}[\text{total number of disagreements in } \widehat{C}_k]\\
\geq&1 - \frac{1}{2}nK(1-\epsilon)^2 \geq 7/8.
\end{aligned}$$

By union bound, the probability that for $k = 1, 2$, any two users in $\widehat{C}_k$ has zero disagreement is at least 3/4.

8.3 Proof of Theorem 3

Consider a compatibility graph with n vertices representing users. Two vertices i, i' are connected if users i, i' have zero disagreement, i.e., $D_{ii'} = 0$. In the noiseless setting, each user cluster forms a clique of size K in the compatibility graph. We call a clique of size K in the compatibility graph a bad clique if it is formed by users from more than one cluster. Then to prove the theorem, it suffices to show that there is no bad clique a.a.s. Since the probability that bad cliques exist increases in ϵ, without loss of generality, we assume $K(1-\epsilon) < 1$.

Recall that B_{kl} is $+1$ or -1 with equal probability. Define $S_k = \{l : B_{kl} = +1\}$ for $k = 1, \dots, r$. As $r \to \infty$, by Chernoff bound, we get that a.a.s., for any $k_1 \neq k_2$

$$|S_{k_1} \Delta S_{k_2}| \triangleq |\{l : B_{k_1 l} \neq B_{k_2 l}\}| \geq \frac{r}{4}. \tag{6}$$

Assume this condition holds throughout the proof.

Fix a set of K users that consists of users from t different clusters. Without loss of generality, assume these users are from cluster $1, \dots, t$. Let n_k denote the number of users from the cluster k and define $n_{\max} = \max_k n_k$. By definition, $2 \leq t \leq t_{\max} \triangleq \min\{r, K\}$, $n_{\max} < K$ and $\sum_{k=1}^t n_k = K$. For any movie j in cluster l, among the K ratings given by these users, there are $\sum_{k=1}^t n_k \mathbb{I}_{\{l \in S_k\}}$ ratings being $+1$ and $\sum_{k=1}^t n_k \mathbb{I}_{\{l \notin S_k\}}$ ratings being -1. Let E_j denote the event that the observed ratings for movie j by these K users are the same. Then,

$$\begin{aligned}
\mathbb{P}[E_j] =&1 - \Big(1 - \epsilon^{\sum_{k=1}^t n_k \mathbb{I}_{\{l \in S_k\}}}\Big)\Big(1 - \epsilon^{\sum_{k=1}^t n_k \mathbb{I}_{\{l \notin S_k\}}}\Big)\\
\leq&\exp\Big(-(1 - \epsilon^{\sum_{k=1}^t n_k \mathbb{I}_{\{l \in S_k\}}})(1 - \epsilon^{\sum_{k=1}^t n_k \mathbb{I}_{\{l \notin S_k\}}})\Big)\\
\leq&\exp\Big(-\frac{1}{4}(1-\epsilon)^2 \sum_{k=1}^t n_k \mathbb{I}_{\{l \in S_k\}} \sum_{k=1}^t n_k \mathbb{I}_{\{l \notin S_k\}}\Big).
\end{aligned}$$

Let $p_{n_1 \dots n_t}$ be the probability that K users, out of which n_k are from cluster k, form a bad clique. Because $\{E_j\}$ are independent and there are K movies in each movie cluster,

$$\begin{aligned}
&p_{n_1 \dots n_t}\\
&= \prod_{j=1}^n \mathbb{P}[E_j]\\
&\leq \exp\Big(-\frac{1}{4}K(1-\epsilon)^2 \sum_{l=1}^r \big(\sum_{k=1}^t n_k \mathbb{I}_{\{l \in S_k\}} \sum_{k=1}^t n_k \mathbb{I}_{\{l \notin S_k\}}\big)\Big)\\
&= \exp\Big(-\frac{1}{4}K(1-\epsilon)^2 \sum_{1 \leq k_1 < k_2 \leq t} n_{k_1} n_{k_2} |S_{k_1} \Delta S_{k_2}|\Big)\\
&\leq \exp\Big(-C_1 n(1-\epsilon)^2 \sum_{k=1}^t n_k(K - n_k)\Big) \qquad (7)
\end{aligned}$$

for some constant C_1. For a large enough constant C in the assumption in the statement of the theorem, we have

$$K\exp(-C_1 n(1-\epsilon)^2(K - n_k)) \leq n^{-3}, \quad n_k \leq \frac{K}{2}, \tag{8}$$

$$K\exp(-C_1 n(1-\epsilon)^2 n_k) \leq n^{-3}, \quad n_k > \frac{K}{2}. \tag{9}$$

Below we show that the probability of bad cliques existing goes to zero. By the Markov inequality and linearity of expectation,

$$\begin{aligned}
&\mathbb{P}[\text{Number of bad cliques} \geq 1]\\
&\leq \mathbb{E}[\text{Number of bad cliques}]\\
&= \sum_{t=2}^{t_{\max}} \binom{r}{t} \sum_{n_1 + \dots + n_t = K} \binom{K}{n_1} \cdots \binom{K}{n_t} p_{n_1 \dots n_t}\\
&= \sum_{t=2}^{t_{\max}} \binom{r}{t} \sum_{n_1 + \dots + n_t = K} \binom{K}{n_1} \cdots \binom{K}{n_t} p_{n_1 \dots n_t}\\
&\big[\mathbb{I}_{\{n_{\max} \leq K/2\}} + \mathbb{I}_{\{n_{\max} > K/2\}}\big]. \qquad (10)
\end{aligned}$$

The first term in (10) is bounded as

$$\begin{aligned}
&\sum_{t=2}^{t_{\max}} \binom{r}{t} \sum_{\substack{n_1 + \dots + n_t = K\\ n_{\max} \leq K/2}} \binom{K}{n_1} \cdots \binom{K}{n_t} p_{n_1 \dots n_t}\\
&\leq \sum_{t=2}^{t_{\max}} r^t \sum_{\substack{n_1 + \dots + n_t = K\\ n_{\max} \leq K/2}} \prod_{k=1}^t (Ke^{-C_1(1-\epsilon)^2(K - n_k)})^{n_k}\\
&\leq \sum_{t=2}^{t_{\max}} r^t K^t n^{-3K} = o(1), \qquad (11)
\end{aligned}$$

where the first inequality follows from the fact that $\binom{K}{n_k} \leq K^{n_k}$ and (7), and the second inequality follows from (8).

The second term in (10) is bounded as

$$\begin{aligned}
&\sum_{t=2}^{t_{\max}} \binom{r}{t} \sum_{\substack{n_1+\cdots+n_t=K\\ n_{\max}>K/2}} \binom{K}{n_1}\cdots\binom{K}{n_t} p_{n_1\ldots n_t}\\
&\le \sum_{t=2}^{t_{\max}} r^t \sum_{\substack{n_1+\cdots+n_t=K\\ n_{\max}>K/2}} (Ke^{-C_1 n(1-\epsilon)^2 n_{\max}})^{K-n_{\max}}\\
&\qquad \prod_{k:n_k<n_{\max}} (Ke^{-C_1(1-\epsilon)^2(K-n_k)})^{n_k}\\
&\le \sum_{t=2}^{t_{\max}} \mathbb{I}_{\{t\le K-n_{\max}+1\}} r^t K^t n^{-6(K-n_{\max})} = o(1), \qquad (12)
\end{aligned}$$

where the first inequality follows from the fact that $\binom{K}{n_k} \le \min\{K^{n_k}, K^{K-n_k}\}$ and (7), and the second inequality follows from (8) and (9) and the fact that $t \le K - n_{\max} + 1$. Therefore we conclude that $\mathbb{P}[\text{Number of bad cliques} \ge 1] = o(1)$.

8.4 Proof of Theorem 4

We first introduce some notations. Let $u_{C,k}$ be the normalized characteristic vector of user cluster k, i.e., $u_{C,k}(i) = 1/\sqrt{K}$ if user i is in cluster k and $u_{C,k}(i) = 0$ otherwise. Thus, $||u_{C,k}||_2 = 1$. Let $U_C = [u_{C,1}, \ldots, u_{C,r}]$. Similarly, let $v_{C,l}$ be the normalized characteristic vector of movie cluster l and $V_C = [v_{C,1}, \ldots, v_{C,r}]$. It is not hard to see that the rating matrix R can be written as $R = KU_C BV_C^\top$. Denote the SVD of the block rating matrix B by $B = U_B\Sigma_B V_B^\top$, then the SVD of R is simply $R = UK\Sigma_B V^\top$, where $U = U_C U_B$ and $V = V_C V_B$. When $r \to \infty$, B has full rank almost surely [7]. We will assume B is full rank in the following proofs, which implies that $U_B U_B^\top = I$ and $V_B V_B^\top = I$. Note that $UU^\top = U_C U_C^\top, VV^\top = V_C V_C^\top$ and $UV^\top = U_C U_B V_B^\top V_C^\top$.

We now briefly recall the subgradient of the nuclear norm [11]. Define T to be the subspace spanned by all matrices of the form $UA^\top$ or $AV^\top$ for any $A \in \mathbb{R}^{n\times r}$. The orthogonal projection of any matrix $M \in \mathbb{R}^{n\times n}$ onto the space T is given by $\mathcal{P}_T(M) = UU^\top M + MVV^\top - UU^\top MVV^\top$. The projection of M onto the complement space $T^\perp$ is $\mathcal{P}_{T^\perp}(M) = M - \mathcal{P}_T(M)$. Then $M \in \mathbb{R}^{n\times n}$ is a subgradient of $||X||_*$ at $X = R$ if and only if $\mathcal{P}_T(M) = UV^\top$ and $||\mathcal{P}_{T^\perp}(M)|| \le 1$.

Proof of Lemma 1. Assume user i is from user cluster k and movie j is in movie cluster l, then

$$|(UV^\top)_{ij}| = |(U_B V_B^\top)_{kl}|/K \le 1/K = r/n,$$

where the inequality follows from the Cauchy-Schwartz inequality. By definition $\mu \le \sqrt{r}$. □

Next we establish the concentration property of $\widehat{R}$. By definition the conditional expectation of $\widehat{R}$ is given by

$$\mathbb{E}[\widehat{R}|R] = (1-\epsilon)(1-2p)R := \bar{R}.$$

Furthermore, the variance is given by

$$\mathrm{Var}[\widehat{R}_{ij}|R] = (1-\epsilon) - (1-\epsilon)^2(1-2p)^2 := \sigma^2.$$

The following corollary applies Theorem 1.4 in [40] to bound the spectral norm $\|\widehat{R} - \bar{R}\|$.

Corollary 1. *If $\sigma^2 \ge C' \log^4 n/n$ for a constant C', then conditioned on R,*

$$\|\widehat{R} - \bar{R}\| \le 3\sigma\sqrt{n} \quad \textit{a.a.s.} \qquad (13)$$

Proof. We adopt the trick called dilations [39]. In particular, define A as

$$A = \begin{bmatrix} \mathbf{0} & \widehat{R} - \mathbb{E}[\widehat{R}|R] \\ \widehat{R}^\top - \mathbb{E}[\widehat{R}^\top|R] & \mathbf{0} \end{bmatrix}. \qquad (14)$$

Observe that $\|A\| = \|\widehat{R} - \mathbb{E}[\widehat{R}|R]\|$, so it is sufficient to prove the theorem for $\|A\|$. Conditioned on R, A is a random symmetric $2n \times 2n$ matrix with each entry bounded by 1, and a_{ij} $(1 \le i < j \le 2n)$ are independent random variables with mean 0 and variance *at most* σ^2. By Theorem 1.4 in [40] , if $\sigma \ge C'n^{-1/2}\log^2 n$, then conditioned on R a.a.s.

$$\begin{aligned}\|\widehat{R} - \mathbb{E}[\widehat{R}|R]\| = \|A\| &\le 2\sigma\sqrt{2n} + C(2\sigma)^{1/2}(2n)^{1/4}\log(2n)\\ &\le 3\sigma\sqrt{n}. \qquad (15)\end{aligned}$$

□

Proof of Theorem 4. For any feasible Y that $Y \ne R$, we have to show that $\Delta(Y) = \langle \widehat{R}, R\rangle - \lambda||R||_* - (\langle \widehat{R}, Y\rangle - \lambda||Y||_*) > 0$. Rewrite $\Delta(Y)$ as

$$\begin{aligned}\Delta(Y) =& \langle \bar{R}, R - Y\rangle + \langle \widehat{R} - \bar{R}, R - Y\rangle\\ &+ \lambda(||Y||_* - ||R||_*). \qquad (16)\end{aligned}$$

The first term in (16) can be written as

$$\begin{aligned}\langle \bar{R}, R - Y\rangle &= (1-\epsilon)(1-2p)\langle R, R - Y\rangle\\ &= (1-\epsilon)(1-2p)||R - Y||_1,\end{aligned}$$

where the second equality follows from the fact that $Y_{ij} \in [-1, 1]$ and $R_{ij} = \mathrm{sgn}(R_{ij})$. Define the normalized noise matrix $W = (\widehat{R} - \bar{R})/\lambda$. Note that $||W||_\infty \le 1/\lambda$ and $\mathrm{Var}(W_{ij}) \le 1/9n$. The second term in (16) becomes $\langle \widehat{R} - \bar{R}, R - Y\rangle = \lambda\langle W, R - Y\rangle$. By Corollary 1, $||W|| \le 1$ almost surely. Thus $UV^\top + \mathcal{P}_{T^\perp}(W)$ is a subgradient of $||X||_*$ at $X = R$. Hence, for the third term in (16), $\lambda(||Y||_* - ||R||_*) \ge \lambda\langle UV^\top + \mathcal{P}_{T^\perp}(W), Y - R\rangle$. Therefore,

$$\begin{aligned}
&\Delta(Y)\\
&\ge (1-\epsilon)(1-2p)||R - Y||_1 + \lambda\langle UV^\top - \mathcal{P}_T(W), Y - R\rangle\\
&\ge [(1-\epsilon)(1-2p) - \lambda(||UV^\top||_\infty + ||\mathcal{P}_T(W)||_\infty)]||R - Y||_1\\
&\ge [(1-\epsilon)(1-2p) - \lambda(\mu\sqrt{r}/n + ||\mathcal{P}_T(W)||_\infty)]||R - Y||_1, \qquad (17)
\end{aligned}$$

where the last inequality follows from definition of the incoherence parameter μ. Below we bound the term $||\mathcal{P}_T(W)||_\infty$. From the definition of $\mathcal{P}_T$ and the fact that $U_B U_B^\top = I$ and $V_B V_B^\top = I$,

$$\begin{aligned}||\mathcal{P}_T(W)||_\infty \le& ||U_C U_C^\top W||_\infty + ||WV_C V_C^\top||_\infty\\ &+ ||U_C U_C^\top W V_C V_C^\top||_\infty.\end{aligned}$$

We bound $||U_C U_C^\top W||_\infty$. To bound the term $(U_C U_C^\top W)_{ij}$, assume user i belongs to user cluster k and let $\mathcal{C}_k$ be the set of users in user cluster k. Recall that $u_{C,k}$ is the normalized characteristic vector of user cluster k. Then

$$(U_C U_C^\top W)_{ij} = (u_{C,k}u_{C,k}^\top W)_{ij} = (1/K)\sum_{i'\in\mathcal{C}_k} W_{i'j},$$

which is the average of K independent random variables. By Bernstein's inequality (stated in the supplementary material), with probability at least $1 - n^{-3}$,

$$\left| \sum_{i' \in \mathcal{C}_k} W_{i'j} \right| \le \sqrt{\frac{2}{3r} \log n} + \frac{2 \log n}{\lambda}.$$

Then $\|U_C U_C^\top W\|_\infty \le \frac{1}{K} \left(\sqrt{\frac{2}{3r} \log n} + \frac{2 \log n}{\lambda} \right)$ with probability at least $1 - n^{-1}$. Similarly we bound $\|W V_C V_C^\top\|_\infty$ and $\|U_C U_C^\top W V_C V_C^\top\|_\infty$. Therefore, with probability at least $1 - 3n^{-1}$,

$$\|\mathcal{P}_T(W)\|_\infty \le \frac{C_1}{K} \left(\sqrt{\frac{\log n}{r}} + \frac{\log n}{\lambda} \right) \le \frac{C_2}{K} \sqrt{\frac{\log n}{r}}, \tag{18}$$

for some constants C_1 and C_2, where the second inequality follows from assumption (4). Substituting (18) into (17) and by assumption (4) again, we conclude that $\Delta(Y) > 0$ a.a.s. □

8.5 Proof of Theorem 5

The proof is divided into three parts. Recall that x_i denotes the i-th row of $Pr(\widehat{R}^{(1)})$. We first show that, for most users, x_i is close to the expected value conditioned on R. Then we show that the clusters output by Step 2 are close to the true clusters. Finally, we show that Step 3 exactly recovers the block rating matrix B and Step 4 exactly recovers clusters.

Define $\bar{R}^{(1)} = \mathbb{E}[\widehat{R}^{(1)} | R] = \frac{1}{2}(1-\epsilon)(1-2p)R$ and let $\bar{x}_i$ be the i-th row of $\bar{R}^{(1)}$. We call user i a *good* user if $\|x_i - \bar{x}_i\|_2 \le \tau/2$ where the threshold $\tau = 12(1-\epsilon)^{1/2} \log n$; otherwise it is called a *bad* user. Let $\mathcal{I}$ denote the set of all good users and $\mathcal{I}^c$ denote the set of all bad users. Define good movies in the same way, and let $\mathcal{J}$ denote the set of all good movies and $\mathcal{J}^c$ denote the set of all bad movies. The following lemma shows that the number of bad users (movies) are bounded by $K \log^{-2} n$.

LEMMA 2. *If $\sigma^2 \ge C' \log^4 n/n$ for a constant C', then a.a.s., $|\mathcal{I}^c| \le K \log^{-2} n$ and $|\mathcal{J}^c| \le K \log^{-2} n$.*

PROOF. Let $(\sigma^{(1)})^2 = \frac{1}{2}(1-\epsilon)$. By Corollary 1, $\|\widehat{R}^{(1)} - \bar{R}^{(1)}\| \le 3\sigma^{(1)}\sqrt{n}$. Note that

$$\begin{aligned} \|P_r(\widehat{R}^{(1)}) - \bar{R}\| &\le \|P_r(\widehat{R}^{(1)}) - \widehat{R}^{(1)}\| + \|\widehat{R}^{(1)} - \bar{R}\| \\ &\le 2\|\widehat{R}^{(1)} - \bar{R}\|, \end{aligned}$$

where the second inequality follows from the definition of $P_r(\widehat{R}^{(1)})$ and the fact that $\bar{R}$ has rank r. Since both $P_r(\widehat{R}^{(1)})$ and $\bar{R}$ have rank r, the matrix $P_r(\widehat{R}^{(1)}) - \bar{R}$ has rank at most $2r$, which implies that

$$\|P_r(\widehat{R}^{(1)}) - \bar{R}\|_F^2 \le 8r\|\widehat{R}^{(1)} - \bar{R}\|^2 \le 72(\sigma^{(1)})^2 nr.$$

As $\sum_{i=1}^n \|x_i - \bar{x}_i\|_2^2 = \|P_r(\widehat{R}^{(1)}) - \bar{R}\|_F^2$, we conclude that there are at most $K \log^{-2} n$ users with

$$\|x_i - \bar{x}_i\|_2 > 6\sqrt{2}\sigma^{(1)} r \log n = \tau/2.$$

Similarly we can prove the result for movies. □

The following proposition upper bounds the set difference between the estimated clusters and the true clusters by $K \log^{-2} n$. Let $C_1^*, \dots, C_r^*$ be the true user clusters and Δ denote the set difference.

PROPOSITION 1. *Assume the assumption of Theorem 5 holds. Step 2 of Algorithm 3 outputs $\{\widehat{C}_k\}_{k=1}^r$ and $\{\widehat{D}_l\}_{1=1}^r$ such that, up to a permutation of cluster indices, a.a.s., $\widehat{C}_k \Delta C_k^* \subset \mathcal{I}^c$ and $\widehat{D}_l \Delta D_l^* \subset \mathcal{J}^c$ for all k, l. It follows that for all k, l,*

$$|\widehat{C}_k \Delta C_k^*| \le \frac{K}{\log^2 n}, \quad |\widehat{D}_l \Delta D_l^*| \le \frac{K}{\log^2 n}. \tag{19}$$

PROOF. It suffices to prove the conclusion for the user clusters. Consider two good users $i, i' \in \mathcal{I}$. If they are from the same cluster, we have $\bar{x}_i = \bar{x}_{i'}$ and

$$\|x_i - x_{i'}\| \le \|x_i - \bar{x}_i\| + \|x_{i'} - \bar{x}_{i'}\| \le \tau, \tag{20}$$

where the last inequality follows from Lemma 2. If they are from different clusters, by (6), we have a.a.s.

$$\begin{aligned} \|\bar{x}_i - \bar{x}_{i'}\|_2^2 =& \frac{1}{4}(1-\epsilon)^2(1-2p)^2\|R_i - R_{i'}\|_2^2 \\ \ge& \frac{1}{4}(1-\epsilon)^2(1-2p)^2 n, \end{aligned}$$

where R_i denotes the i-th row of R. Thus,

$$\begin{aligned} \|x_i - x_{i'}\| &\ge \|\bar{x}_i - \bar{x}_{i'}\| - \|x_i - \bar{x}_i\| - \|x_{i'} - \bar{x}_{i'}\| \\ &\ge \frac{1}{2}(1-\epsilon)(1-2p)\sqrt{n} - \tau > \tau, \end{aligned} \tag{21}$$

where the last inequality follows from the assumption (5). Therefore, in the clustering procedure of Step 2, if we choose a good initial user at some iteration, the corresponding estimated cluster will contain all the good users from the same cluster as the initial user and no good user from other clusters. It is not hard to see that the probability of the event that we choose a good initial user in every iteration is lower bounded by

$$\begin{aligned} &\left(1 - \frac{1}{r \log^2 n}\right)\left(1 - \frac{1}{(r-1)\log^2 n}\right) \cdots \left(1 - \frac{1}{\log^2 n}\right) \\ &\ge 1 - \frac{1}{\log^2 n}\left(\frac{1}{r} + \frac{1}{r-1} + \cdots + 1\right) \\ &\ge 1 - \frac{\log r}{\log^2 n} \ge 1 - \frac{1}{\log n}. \end{aligned}$$

Assume the above event holds. Under proper permutation, the initial good user in the k-th iteration is from cluster C_k^* for all k. By the above argument, the set difference $\widehat{C}_k \Delta C_k^* \subset \mathcal{I}^c$. By Lemma 2, (19) follows. □

PROOF OF THEOREM 5. We first show that Step 3 of Algorithm 3 exactly recovers the block rating matrix B. Let V_{kl} denote the total vote that the true user cluster k gives to the true movie cluster l, i.e.,

$$V_{kl} = \sum_{i \in C_k^*} \sum_{j \in D_l^*} \widehat{R}_{ij}^{(2)}.$$

Then by definition of $\widehat{V}_{kl}$,

$$\begin{aligned} |\widehat{V}_{kl} - V_{kl}| \le& \sum_{i \in C_k^* \Delta \widehat{C}_k} \sum_{j \in D_l^* \cup \widehat{D}_l} \mathbb{I}_{\{(i,j) \in \Omega_2\}} \\ &+ \sum_{i \in C_k^* \cup \widehat{C}_k} \sum_{j \in D_l^* \Delta \widehat{D}_l} \mathbb{I}_{\{(i,j) \in \Omega_2\}}. \end{aligned} \tag{22}$$

Without loss of generality, assume $B_{kl} = 1$. By Bernstein inequality and assumption (5), $V_{kl} \ge \frac{1}{4}(1-\epsilon)(1-2p)K^2$

a.a.s. On the other hand, as Ω_2 and $\widehat{R}^{(1)}$ are independent, Ω_2 is independent from $\{\widehat{C}_k\}$ and $\{\widehat{D}_l\}$. It follows from (19) and the Chernoff bound that each term on the right hand side of (22) is upper bounded by $(1-\epsilon)K^2 \log^{-2} n$ a.a.s. Hence, when assumption (5) holds for some large enough constant C, we have $\widehat{V}_{kl} > 0$ thus $\widehat{B}_{kl} = B_{kl}$.

Next we prove that Step 4 clusters the users and movies correctly. Without loss of generality, we only prove the correctness for users. Suppose user i is from cluster k. Recall that R_i denotes the i-th row of R. When $\widehat{B} = B$, we have $\mu_{kj} = R_{ij}$ for $j \in \mathcal{J}$ by definition and Proposition 1. Then

$$\begin{aligned}\langle \widehat{R}_i^{(2)}, \mu_k \rangle =& \langle \widehat{R}_i^{(2)}, R_i \rangle + \langle \widehat{R}_i^{(2)}, \mu_k - R_i \rangle \\ \geq& \langle \widehat{R}_i^{(2)}, R_i \rangle - 2 \sum_{j \in \mathcal{J}^c} |\widehat{R}_{ij}^{(2)}|. \end{aligned} \tag{23}$$

Similarly, for some user i' from cluster $k' \neq k$,

$$\begin{aligned}\langle \widehat{R}_i^{(2)}, \mu_{k'} \rangle =& \langle \widehat{R}_i^{(2)}, R_{i'} \rangle + \langle \widehat{R}_i^{(2)}, \mu_{k'} - R_{i'} \rangle \\ \leq& \langle \widehat{R}_i^{(2)}, R_{i'} \rangle + 2 \sum_{j \in \mathcal{J}^c} |\widehat{R}_{ij}^{(2)}| \end{aligned} \tag{24}$$

For ease of notation, let $t := \frac{1}{2}(1-\epsilon)(1-2p)n$ and $(\sigma^{(2)})^2 = \frac{1}{2}(1-\epsilon)$. By (6), $\langle R_i, R_{i'} \rangle \leq n/2$ for all $i \neq i'$. Then conditioned on R, we have $\mathbb{E}[\langle \widehat{R}_i^{(2)}, R_i \rangle] = t$ and $\mathrm{Var}[\langle \widehat{R}_i^{(2)}, R_i \rangle] \leq n(\sigma^{(2)})^2$, and

$$\mathbb{E}[\langle \widehat{R}_i^{(2)}, R_{i'} \rangle] = \frac{1}{2}(1-\epsilon)(1-2p)\langle R_i, R_i' \rangle \leq t/2$$

and $\mathrm{Var}[\langle \widehat{R}_i^{(2)}, R_{i'} \rangle] \leq n(\sigma^{(2)})^2$. Now by the Bernstein inequality and assumption (5), we have that conditioned on R, a.a.s. $\langle \widehat{R}_i^{(2)}, R_i \rangle > 7t/8$ and $\langle \widehat{R}_i^{(2)}, R_{i'} \rangle < 5t/8$ for all $i \neq i'$.

On the other hand, because $\mathcal{J}$ and Ω_2 are independent, by the Chernoff bound, a.a.s. $\sum_{j \in \mathcal{J}^c} |\widehat{R}_{ij}^{(2)}|$ is upper bounded by $(1-\epsilon)K \log^{-2} n < t/16$ for all i, when assumption (5) holds for some large enough constant C.

Therefore, from (23) and (24), $\langle \widehat{R}_i^{(2)}, \mu_k \rangle > \langle \widehat{R}_i^{(2)}, \mu_{k'} \rangle$ for all $k' \neq k$. □

9. NUMERICAL EXPERIMENTS

In this section, we illustrate the performance of the convex method and the spectral method using synthetic data.

9.1 Convex method

The convex program (2) can be formulated as a semidefinite program (SDP) and solved using a general purpose SDP solver. However this method does not scale well for our problem when the matrix dimension n is large. Instead we apply the accelerated gradient descent method proposed in [25, 37] which aims to solve the optimization problem

$$\min_Y f(Y) + \lambda ||Y||_*$$

for some smooth function $f(Y)$. In our case, the smooth function is linear, i.e., $f(Y) = -\langle \hat{R}, Y \rangle$. Define proximal regularization of $f(Y)$ at X as

$$\begin{aligned} P_\mu(X, Y) =& f(X) - \langle Y - X, \hat{R} \rangle + \frac{\mu}{2} ||Y - X||_F^2 \\ =& - \langle Y, \hat{R} \rangle + \frac{\mu}{2} ||Y - X||_F^2. \end{aligned}$$

for some constant $\mu > 0$. Then it is shown in [25] that (2) is solved by the following iterative algorithm:

$$Y_k = \arg \min_{Y_{ij} \in [-1,1]} P_\mu(Y_{k-1}, Y) + \lambda ||Y||_*. \tag{25}$$

We approximate Y_k by first solving the unconstrained optimization problem

$$\min_Y P_\mu(Y_{k-1}, Y) + \lambda ||Y||_*. \tag{26}$$

and then project each entry of the solution to $[-1, 1]$, where we use $P_{[-1,1]}$ to denote the projection operator. The minimizer of (26) can be explicitly written in terms of the soft-thresholding operator D defined as follows. For any $\gamma \geq 0$ and for any matrix X with SVD $X = U \Sigma V^\top$ where $\Sigma = \mathrm{diag}(\{\sigma_i\})$, define

$$D_\gamma(X) = U \mathrm{diag}(\{\max(\sigma_i - \gamma, 0)\}) V^\top.$$

Intuitively, the soft-thresholding operator D shrinks the singular values of X towards zero. Applying Theorem 2.1 in [9], we get

$$D_{\frac{\lambda}{\mu}} \left(X + \frac{\widehat{R}}{\mu} \right) = \arg \min_Y P_\mu(X, Y) + \lambda ||Y||_*.$$

Thus, the update equation (25) of Y_k is approximated by

$$Y_k = P_{[-1,1]} \left(D_{\frac{\lambda}{\mu}} (Y_{k-1} + \frac{\widehat{R}}{\mu}) \right).$$

This iterative algorithm can further be accelerated to achieve the optimal convergence rate of $O(1/k^2)$, which results Algorithm 4 [25]. Note that we do not use a fixed regularization parameter λ. The algorithm has better performance when we start with $\lambda_0 = 3\sqrt{(1-\epsilon)n}$ as in Theorem 4 and decrease it gradually until it reaches $\bar{\lambda} = \sqrt{(1-\epsilon)n}$. In the experiment, we choose $\mu_k = 1$.

Algorithm 4 Accelerated Gradient Descent Algorithm

```
Input: $\widehat{R}$
Initialization: Set $Y_0 = Y_{-1} = 0$ and $\alpha_0 = \alpha_{-1} = 1$. Pick $\lambda_0 = 3\sqrt{(1-\epsilon)n}$ and $\bar{\lambda} = \sqrt{(1-\epsilon)n}$. Set $\gamma = 0.95$. Set $\mu_k = 1$.
for $k = 0, 1, 2, \ldots$ do
    $Z_k = Y_k + \frac{\alpha_{k-1} - 1}{\alpha_k}(Y_k - Y_{k-1})$
    $Y_{k+1} = P_{[-1,1]}\left(D_{\frac{\lambda_k}{\mu_k}}\left(Z_k + \frac{\hat{R}}{\mu_k}\right)\right)$
    $\alpha_{k+1} = \frac{1 + \sqrt{1 + 4\alpha_k^2}}{2}$
    $\lambda_{k+1} = \max\{\gamma \lambda_k, \bar{\lambda}\}$.
end for
```

We simulate Algorithm 4 on the synthetic data. Assume K and ϵ take the form given by

$$K = n^\beta, \quad \epsilon = 1 - n^{-\alpha}. \tag{27}$$

Theorem 4 shows that the convex program (4) recovers the rating matrix exactly when $\alpha < \beta$, assuming Conjecture 1 holds.

We generate the observed data matrix with $n = 2048$, $p = 0.05$ and various choices of $\beta, \alpha \in (0, 1)$, and apply Algorithm 4. The solution $\widehat{Y}$ is evaluated by the fraction of entries with correct signs, i.e., $\frac{1}{n^2}|\{(i,j) : \mathrm{sign}(\widehat{Y}_{ij}) = R_{ij}\}|$.

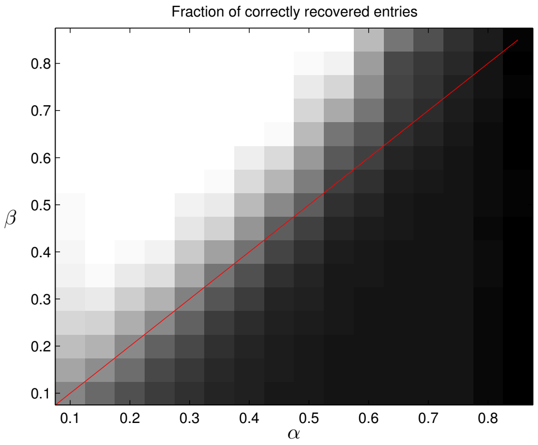

Figure 3: Simulation result of the convex method given in Algorithm 4 with $n = 2048$ and $p = 0.05$. The x-axis corresponds to erasure probability $\epsilon = 1 - n^{-\alpha}$ and y-axis corresponds to cluster size $K = n^{\beta}$. The grey scale of each area represents the fraction of entries with correct signs, with white representing exact recovery and black representing around 50% recovery. The red line shows the performance of the convex method predicted by Theorem 4.

The result is plotted in grey scale in Figure 3. In particular, the white area represents exact recovery and the black area represents around 50% recovery, which is equivalent to random guess. The red line represents $\alpha = \beta$, which shows the performance guarantee given by Theorem 4. As we can see, the simulation results roughly match the theoretical performance guarantee.

9.2 Spectral Method

We simulate the spectral method given in Algorithm 3 on synthetic data. Assume K and ϵ take the form of (27). Theorem 5 shows that the spectral method exactly recovers the clusters when $\alpha < \frac{1}{2}(\beta + 1)$.

We generate the observed data matrix according to our model with $n = 2^{11}, 2^{12}, 2^{13}$ and $p = 0.05$, and various choices of $\beta, \alpha \in (0, 1)$. We apply Algorithm 3 with slight modifications. Firstly, we do not split the observation as in Step 1 but use all the observations for the later steps, i.e., $\Omega_1 = \Omega_2 = \Omega$. Secondly, in Step 2 we use the more robust k-means algorithm to cluster users and movies instead of the thresholding based clustering algorithm. The clustering error is measured by the fraction of mis-clustered users and movies. We say the algorithm succeeds if the clustering error is less than 5%.

For each β, we run the algorithm for several values of α and record the largest α for which the algorithm succeeds. The result is depicted in Fig 4. The solid blue line represents $\alpha = \frac{1}{2}(\beta + 1)$, which shows the performance guarantee of the spectral method given by Theorem 5. The solid red line represents $\alpha = \beta$, which shows the performance guarantee of the convex method given by Theorem 4. We can see that the simulation results of the spectral method are better than its theoretical performance guarantee, but worse than the theoretical performance guarantee of the convex method.

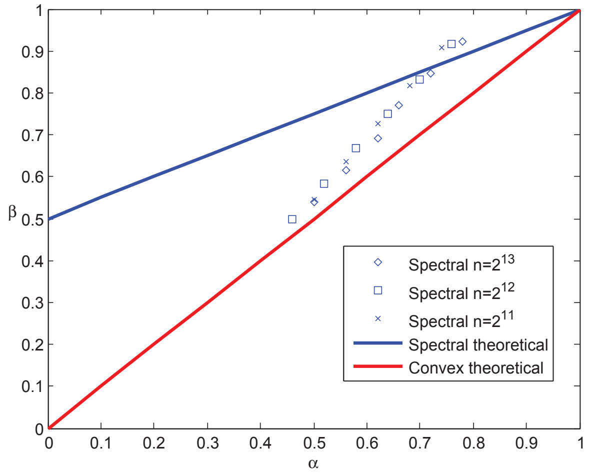

Figure 4: Simulation result of the spectral method given in Algorithm 3 with $n = 2^{11}, 2^{12}, 2^{13}$ and $p = 0.05$. The x-axis corresponds to erasure probability $\epsilon = 1 - n^{-\alpha}$ and y-axis corresponds to cluster size $K = n^{\beta}$. Each data point in the plot indicates the maximum value of α for which the spectral method succeeds with a given β. The blue solid line shows the performance of the spectral method predicted by Theorem 5. The red solid line shows the performance of the convex method predicted by Theorem 4.

10. CONCLUDING REMARKS

This paper studies the problem of inferring hidden row and column clusters of binary matrices from a few noisy observations through theoretical analysis and numerical experiments. More extensive simulation results will be presented in a longer version of the paper. Several future directions are of interest. First, proving Conjecture 1 or removing the dependency on incoherence μ [15] is important to fully understand the performance of the convex method. Second, a tight performance analysis of the ML estimation (1) is needed to achieve the lower bound. Third, it is interesting to extend our analysis to block rating matrices having real-valued entries.

11. ACKNOWLEDGMENTS

Research supported in part by AFOSR MURI FA 9550-10-1-0573 and NSF ECCS 10-28464.

12. REFERENCES

[1] S. T. Aditya, O. Dabeer, and B. Dey. A channel coding perspective of collaborative filtering. *IEEE Transactions on Information Theory*, 57(4):2327–2341, 2011.

[2] E. Arias-Castro and N. Verzelen. Community detection in random networks. *arXiv preprint arXiv:1302.7099*, 2013.

[3] S. Balakrishnan, M. Kolar, A. Rinalso, and A. Singh. Statistical and computational tradeoffs in biclustering. In *NIPS 2011 Workshop on Computational Trade-offs in Statistical Learning*, 2011.

[4] K. Barman and O. Dabeer. Analysis of a collaborative filter based on popularity amongst neighbors. *IEEE*

Transactions on Information Theory, 58(12):7110–7134, 2012.

[5] Q. Berthet and P. Rigollet. Complexity theoretic lower bounds for sparse principal component detection. *J. Mach. Learn. Res.*, 30:1046–1066 (electronic), 2013.

[6] Q. Berthet and P. Rigollet. Optimal detection of sparse principal components in high dimension. *Ann. Statist.*, 41(1):1780–1815, 2013.

[7] J. Bourgain, V. H. Vu, and P. M. Wood. On the singularity probability of discrete random matrices. *Journal of Functional Analysis*, 258(2):559–603, 2010.

[8] S. Busygin, O. Prokopyev, and P. M. Pardalos. Biclustering in data mining. *Computers and Operations Research*, 35(9):2964 – 2987, 2008.

[9] J.-F. Cai, E. J. Candès, and Z. Shen. A singular value thresholding algorithm for matrix completion. *SIAM J. on Optimization*, 20(4):1956–1982, Mar. 2010.

[10] E. J. Candès, X. Li, Y. Ma, and J. Wright. Robust principal component analysis? *J. ACM*, 58(3):11:1–11:37, June 2011.

[11] E. J. Candès and B. Recht. Exact matrix completion via convex optimization. *Found. Comput. Math.*, 9(6):717–772, Dec. 2009.

[12] E. J. Candès and T. Tao. The power of convex relaxation: near-optimal matrix completion. *IEEE Trans. Inf. Theor.*, 56(5):2053–2080, May 2010.

[13] V. Chandrasekaran and M. I. Jordan. Computational and statistical tradeoffs via convex relaxation. *PNAS*, 110(13):E1181–E1190, 2013.

[14] V. Chandrasekaran, S. Sanghavi, P. Parrilo, and A. Willsky. Rank-sparsity incoherence for matrix decomposition. *SIAM Journal on Optimization*, 21(2):572–596, 2011.

[15] Y. Chen. Incoherence-optimal matrix completion. 2013, available at http://arxiv.org/abs/1310.0154.

[16] Y. Chen, A. Jalali, S. Sanghavi, and C. Caramanis. Low-rank matrix recovery from errors and erasures. *IEEE Transactions on Information Theory*, 59(7):4324–4337, 2013.

[17] Y. Chen, A. Jalali, S. Sanghavi, and H. Xu. Clustering partially observed graphs via convex optimization. In *ICML*, 2011, available at: http://arxiv.org/abs/1104.4803.

[18] Y. Chen, S. Sanghavi, and H. Xu. Clustering sparse graphs. In *NIPS*, 2012, available at: http://arxiv.org/abs/1210.3335.

[19] Y. Chen and J. Xu. Statistical-computational tradeoffs in planted problems and submatrix localization with a growing number of clusters and submatrices. 2014, available at http://arxiv.org/abs/1402.1267.

[20] Y. Cheng and G. M. Church. Biclustering of expression data. *Proceedings of International Conference on Intelligent Systems for Molecular Biology (ISMB)*, 8:93–103, 2000.

[21] M. A. Davenport, Y. Plan, E. van den Berg, and M. Wootters. 1-bit matrix completion. Sept. 2012, available at http://arxiv.org/abs/1209.3672.

[22] S. Fortunato. Community detection in graphs. *Physics Reports*, 486(3-5):75 – 174, 2010.

[23] J. A. Hartigan. Direct clustering of a data matrix. *Journal of the American Statistical Association*, 67(337):123–129, 1972.

[24] P. Jain, P. Netrapalli, and S. Sanghavi. Low-rank matrix completion using alternating minimization. In *Proceedings of the 45th Annual ACM Symposium on Symposium on Theory of Computing*, STOC '13, pages 665–674, New York, NY, USA, 2013. ACM.

[25] S. Ji and J. Ye. An accelerated gradient method for trace norm minimization. In *Proceedings of the 26th Annual International Conference on Machine Learning*, ICML '09, pages 457–464, New York, NY, USA, 2009. ACM.

[26] D. Jiang, C. Tang, and A. Zhang. Cluster analysis for gene expression data: A survey. *IEEE Trans. on Knowl. and Data Eng.*, 16(11):1370–1386, Nov. 2004.

[27] R. Kannan and S. Vempala. Spectral algorithms. *Found. Trends Theor. Comput. Sci.*, 4, Mar 2009.

[28] M. Kolar, S. Balakrishnan, A. Rinaldo, and A. Singh. Minimax localization of structural information in large noisy matrices. In *NIPS*, 2011.

[29] Y. Koren, R. Bell, and C. Volinsky. Matrix factorization techniques for recommender systems. *Computer*, 42(8):30–37, 2009.

[30] R. Krauthgamer, B. Nadler, and D. Vilenchik. Do semidefinite relaxations really solve sparse pca? 2013, available at http://arxiv.org/abs/1306.3690.

[31] G. Linden, B. Smith, and J. York. Amazon.com recommendations: item-to-item collaborative filtering. *Internet Computing, IEEE*, 7(1):76–80, 2003.

[32] Z. Ma and Y. Wu. Computational barriers in minimax submatrix detection. 2013, available at http://arxiv.org/abs/1309.5914.

[33] S. C. Madeira and A. L. Oliveira. Biclustering algorithms for biological data analysis: A survey. *IEEE/ACM Trans. Comput. Biol. Bioinformatics*, 1(1):24–45, Jan. 2004.

[34] F. McSherry. Spectral partitioning of random graphs. In *42nd IEEE Symposium on Foundations of Computer Science*, pages 529 – 537, Oct. 2001.

[35] B. Recht. A simpler approach to matrix completion. *J. Mach. Learn. Res.*, 12:3413–3430, Dec 2011.

[36] K. Rohe, S. Chatterjee, and B. Yu. Spectral clustering and the high-dimensional stochastic blockmodel. *The Annals of Statistics*, 39(4):1878–1915, 2011.

[37] K.-C. Toh and S. Yun. An accelerated proximal gradient algorithm for nuclear norm regularized linear least squares problems. *Pacific Journal of Optimization*, 2010.

[38] D.-C. Tomozei and L. Massoulié. Distributed user profiling via spectral methods. *SIGMETRICS Perform. Eval. Rev.*, 38(1):383–384, June 2010.

[39] J. Tropp. User-friendly tail bounds for sums of random matrices. *Foundations of Computational Mathematics*, 12(4):389–434, 2012.

[40] V. H. Vu. Spectral norm of random matrices. *Combinatorica*, 27(6):721–736, 2007.

[41] S.-Y. Yun and A. Proutiere. Community detection via random and adaptive sampling. 2014, available at http://arxiv.org/abs/1402.3072.