MC2G: An Efficient Algorithm for Matrix Completion With Social and Item Similarity Graphs

Qiaosheng Zhang, Geewon Suh, *Graduate Student Member, IEEE*, Changho Suh, *Senior Member, IEEE*, and Vincent Y. F. Tan, *Senior Member, IEEE*

***Abstract*—In this paper, we design and analyze MC2G (Matrix Completion with 2 Graphs), an efficient algorithm that performs matrix completion in the presence of social and item similarity graphs. MC2G runs in quasilinear time and is parameter free. It is based on spectral clustering and local refinement steps. For the matrix completion problem which possesses additional block structures in its rows and columns, we derive the expected number of sampled entries required for MC2G to succeed, and further show that it matches an information-theoretic lower bound up to a constant factor for a wide range of parameters. We perform extensive experiments on both synthetic datasets and a semi-real dataset inspired by real graphs. The experimental results show that MC2G outperforms other state-of-the-art matrix completion algorithms.**

***Index Terms*—Matrix completion, community detection, stochastic block model, graph side information.**

I. Introduction

With the ubiquity of social networks such as Facebook and Twitter, it is increasingly convenient to collect similarity information amongst users. It has been shown that exploiting this similarity information in the form of a *social graph* can significantly improve the quality of recommender systems [1]–[8] compared to traditional recommendation algorithms (e.g., collaborative filtering [9]) that rely merely on rating information. This improvement is particularly pronounced in the presence of the so-called *cold-start problem* in which we would like to recommend items to a user who has not rated any items, but we do possess his/her similarity information with other users. Similarly, an *item similarity graph* is sometimes also available for exploitation—it can be constructed either from the features of items [10], or from users' behavior history (as has been done by Taobao [11]). Again, this can help in solving the *dual cold-start problem*, namely the learner has no information about new items that have not been rated by any user.

While there have been numerous studies considering how to exploit graph side information to enhance recommender systems, most of the algorithms developed so far exploit *only one* graph (either the social or the item similarity graph). As mentioned above, *both* graphs are often available in many real-life applications, and it has been shown in a prior theoretical study [12] that there are scenarios in which exploiting two graphs yields strictly more benefits than exploiting only one graph. This work builds upon [12] which focuses on *fundamental limits*, but does not propose computationally efficient algorithms that achieve the limits. Our main contribution is to design and analyze a computationally efficient algorithm—which we name MC2G—for a matrix completion problem, wherein both the social and item similarity graphs are available. Our algorithm MC2G is motivated by the observation that users and items in real recommender systems often share similarities and are clustered. The key idea behind MC2G is to first find the clusters of users and items as accurately as possible, and then predict each missing entry based on other available entries that belong to the same clusters of users and items. Thus, MC2G can be applied to modern recommender systems that contain side information in the form of social and item similarity graphs. On the matrix completion problem described below, we provide theoretical guarantees on the expected number of sampled entries for MC2G to succeed, and further show that it meets an information-theoretic lower bound up to a constant factor.

We consider a matrix completion problem in which there are n users and m items. Users are partitioned into $k_1 \geq 2$ clusters, while items are partitioned into $k_2 \geq 2$ clusters. Users' ratings to items are chosen from an *arbitrarily* pre-assigned finite set (e.g., $\{1, 2, 3, 4, 5\}$ for Netflix prize challenge [13]). The $n \times m$ rating matrix is generated according to a generative model which we describe in Section II. The learner observes three pieces of information: (i) a sub-sampled rating matrix with each entry being sampled independently with probability p; (ii) a social graph generated according to a celebrated generative model for random graphs—the *stochastic block model* (SBM) [14]; and (iii) an item similarity graph generated according to another

Manuscript received June 5, 2021; revised January 1, 2022 and April 18, 2022; accepted May 3, 2022. Date of publication May 11, 2022; date of current version June 7, 2022. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. David I. Shuman. The work of Qiaosheng Zhang and Vincent Y. F. Tan was supported by the Singapore National Research Foundation Fellowship under Grant A-0005077-01-00. The work of Geewon Suh and Changho Suh was supported by the Institute for Information & Communications Technology Planning & Evaluation grant funded by the Korea government (MSIT) under Grant 2020-0-00626. *(Qiaosheng Zhang and Geewon Suh contributed equally to this work.) (Corresponding author: Qiaosheng Zhang.)*

Qiaosheng Zhang is with Shanghai Artificial Intelligence Laboratory, Shanghai 200232, China (e-mail: zhangqiaosheng@pjlab.org.cn).

Geewon Suh and Changho Suh are with the School of Electrical Engineering, Korea Advanced Institute of Science and Technology, Daejeon 34141, South Korea (e-mail: gwsuh91@kaist.ac.kr; chsuh@kaist.ac.kr).

Vincent Y. F. Tan is with the Department of Electrical and Computer Engineering and Department of Mathematics, National University of Singapore, Singapore 117583 (e-mail: vtan@nus.edu.sg).

This article has supplementary downloadable material available at https://doi.org/10.1109/TSP.2022.3174423, provided by the authors.

Digital Object Identifier 10.1109/TSP.2022.3174423

SBM. The task is to exactly recover the clusters of both users and items, as well as to complete a matrix that represents the nominal ratings from certain user clusters to certain item clusters (called *nominal matrix*). Essentially, our problem is about exactly recovering a matrix with hidden block structures in the presence of noise and graph side information. Our model significantly generalizes the models considered in related works with theoretical guarantees [1], [2], [12], by relaxing some constraints therein, e.g., (i) users/items are only partitioned into two equal-sized clusters, and (ii) only binary ratings are allowed.

A. Main Contributions

Our main contributions are summarized as follows.

1) We develop a computationally efficient algorithm MC2G, which is a multi-stage algorithm that follows the "*from global to local*" principle. It first adopts a *spectral clustering method* on graphs to obtain initial estimates of user/item clusters, and then refines each user/item individually based on *local* maximum likelihood estimation (MLE). MC2G is also a parameter-free algorithm that does not need the knowledge of the model parameters. Under the symmetric setting described in Section IV-A, we show that MC2G exactly recover the user/item clusters as well as the nominal matrix with high probability as long as the number of samples exceeds a bound presented in Theorem 1.
2) We also provide an information-theoretic lower bound that matches the bound in Theorem 1 up to a constant factor; this demonstrates the order-wise optimality of MC2G. As a by-product, the aforementioned theoretical results also generalize the theory developed in the prior work [12], which was focused on a simpler setting in which both users and items are partitioned into two clusters.
3) We conduct extensive experiments on synthetic datasets to verify that the results show keen agreement with the derived theoretical guarantee of MC2G in Theorem 1. We further demonstrate the superior performance of MC2G by comparing it with several state-of-the-art matrix completion algorithms, such as OPTSPACE [15], SoRec [3], and a spectral clustering method with local refinements using *only* the social graph or only the item graph [1].
4) Finally, we assess MC2G on a semi-real dataset consisting of real graphs but synthetic ratings. We adopt the LastFM social network [16] and political blogs network [17] as the social and item similarity graphs, respectively. Our experimental results show that MC2G works well under the real graphs; this further confirms that MC2G is universal, as the real graphs do not satisfy the symmetry assumptions. Finally, we show that MC2G outperforms the other aforementioned matrix completion algorithms on this semi-real dataset.

B. Related Works

Due to the wide applicability of matrix completion (such as recommender systems), the past decade has witnessed the developments of many efficient matrix completion algorithms, such as [18]–[23]. In the context of recommender systems, the design of algorithms that exploit graph side information (especially the social graph) has attracted much attention. For example, researchers have proposed a variety of matrix factorization-based algorithms that incorporate graph side information [3]–[8], [24], [25], where the graph is used to design additional regularization terms or modify the existing matrix model. Another popular approach is via the neighborhood-based algorithms [26], [27], where users' ratings are predicted based on the ratings of their neighbors, and the neighborhood is defined using social graphs. Recently, some deep learning-based algorithms [28]–[30] (relying on convolutional neural networks) have also been proposed for recommender systems with graph side information. Among the aforementioned works, [6]–[8], [28], [30] also considered simultaneously exploiting both the social and item similarity graphs in their algorithms. For example, [7] proposed an algorithm, called kernelized probabilistic matrix factorization (KPMF), that effectively incorporates the two graphs into the matrix factorization process. Rao *et al.* [8] incorporated the two graphs as regularization terms in the matrix completion problem, and developed a scalable algorithm based on efficient Hessian-vector multiplication schemes. Although these algorithms usually yield better empirical performance, most of them do not provide any theoretical guarantee. Notably, [8] performed a theoretical analysis when the observed entries are perturbed by noise, and provided an upper bound on the estimation error between the true matrix and recovered matrix. However, it remains unexplored in [8] that (theoretically) by how much one can improve the performance with the aid of graphs, while this work quantifies the gains of exploiting social and item similarity graphs. In Section IV-D, we provide a more detailed comparison about the theoretical results of this work, ref. [8], and other related works on matrix completion.

We note that another line of works focused on the fundamental limits of matrix completion in which the matrix is generated according to a certain generative model for the clusterings of the users and/or items. Ahn *et al.* [1] considered a setting where ratings are binary and a graph encodes the structure of two clusters, and characterized the expected number of sampled entries required for matrix completion. Follow-up works [2], [31] relaxed the assumptions in [1], but are still restricted to exploiting the use of a single graph. The recent work [12] investigated a more general setting in which both the social and item similarity graphs are available, and quantified the gains of the two graphs via information-theoretic lower and upper bounds. However, a computationally efficient algorithm that achieves the limit promised by MLE was not developed in [12]. Given that the MLE is not computationally feasible, there is a pressing need to develop efficient algorithms. This precisely sets the goal of this work. Additionally, this work studies a generalized setting that spans multiple user/item clusters and discrete-valued rating matrices. This is in contrast to [12] which focuses on two clusters and binary ratings. Finally, it is worth mentioning that our algorithm MC2G is universally applicable to all matrix completion problems with two-sided graph side information, i.e., it is not restricted to the setting in [12] or the setting in this work.

TABLE I
NOMINAL RATINGS FROM USERS TO ITEMS

	Cluster $\mathcal{I}_1$	Cluster $\mathcal{I}_2$	.....	Cluster $\mathcal{I}_{k_2}$
Cluster $\mathcal{U}_1$	z_{11}	z_{12}	...	z_{1k_2}
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$
Cluster $\mathcal{U}_{k_1}$	$z_{k_1 1}$	$z_{k_1 2}$	...	$z_{k_1 k_2}$

Another field relevant to this work is *community detection*, which is the problem of partitioning nodes of an undirected graph into different clusters/communities. When the graphs are generated according to SBMs, the information-theoretic limits for exact recovery of clusters [32]–[37] have been established. These limits also play a role in establishing the theoretical guarantee of MC2G (see the third item in Remark 5), as our algorithm includes the clustering step for users and items in the process of matrix completion. It has also been shown that side information is in general helpful for community detection [38]–[41]. Besides, our problem is also related to the labelled/weighted SBM problem, if the two SBMs that govern the social and item similarity graphs are merged to a single unified SBM (interested readers are referred to [12, Remark 4] for details).

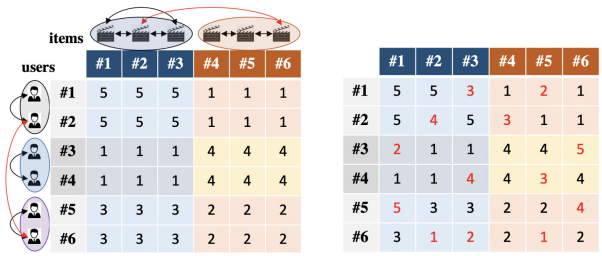
C. Outline

This paper is organized as follows. We first introduce the problem setup in Section II, and then describe our efficient algorithm MC2G in Section III. Section IV presents our main theoretical results: (i) the theoretical guarantee for MC2G and (ii) the information-theoretical lower bound. These results are proved in Sections V and VI, respectively. Experimental results are presented in Section VII.

II. PROBLEM STATEMENT

We consider a recommender system with n users and m items. Ratings from users to items are chosen from an arbitrary finite alphabet $\mathcal{Z}$ (e.g., $\mathcal{Z} = \{1, 2, 3, 4, 5\}$). It is assumed that users are partitioned into $k_1 \geq 2$ disjoint clusters $\{\mathcal{U}_1, \mathcal{U}_2, \ldots, \mathcal{U}_{k_1}\}$, and items are partitioned into $k_2 \geq 2$ disjoint clusters $\{\mathcal{I}_1, \mathcal{I}_2, \ldots, \mathcal{I}_{k_2}\}$. We define[1] $\sigma : [n] \to [k_1]$ as the *label function for users* such that $\sigma(i) = a$ if user i belongs to cluster $\mathcal{U}_a$. On the contrary, each clutser $\mathcal{U}_a$ can be represented as $\mathcal{U}_a = \{i \in [n] : \sigma(i) = a\}$. Thus, σ can be viewed as an alternative (and more concise) representation of the clusterings of users $\{\mathcal{U}_a\}_{a \in [k_1]}$. Similarly, we define $\tau : [m] \to [k_2]$ as the *label function for items* such that $\tau(j) = b$ if item j belongs to cluster $\mathcal{I}_b$.

As users in the same cluster are more likely to share similar preference (which is called *homophily* [42] in the social sciences), we introduce the notion of *nominal ratings* to represent the levels of interest from certain user clusters to certain item clusters. Specifically, for all the users in cluster $\mathcal{U}_a$, their nominal ratings to all the items in cluster $\mathcal{I}_b$ (where $a \in [k_1]$, $b \in [k_2]$) are given by $z_{ab} \in \mathcal{Z}$ (as shown in Table I). That is, the nominal ratings given by users in the same clusters are the same, and the nominal ratings received by items in the same cluster are also identical. Thus, given σ (the labels of n users), τ (the labels of m items), and $\{z_{ab}\}$ (the nominal ratings), the corresponding *nominal matrix* $\mathsf{N} \in \mathcal{Z}^{n \times m}$ is an $n \times m$ matrix that contains the nominal ratings from n users to m items, and each entry N_{ij} (the nominal rating from user i to item j) equals $z_{\sigma(i)\tau(j)}$. An example of the nominal matrix is illustrated in Fig. 1(a).



(a) *Nominal matrix* N (b) *Personalized rating matrix*

Fig. 1. An example with 6 users (partitioned into 3 clusters) and 6 items (partitioned into 2 clusters). The nominal ratings are chosen from $\mathcal{Z} \in \{1, 2, 3, 4, 5\}$, and are set to be $z_{11} = 5, z_{12} = 1, z_{21} = 1, z_{22} = 4, z_{31} = 3, z_{32} = 2$.

We assume the *personalized rating* $V_{ij} \in \mathcal{Z}$ of user i to item j is a stochastic function of the nominal rating N_{ij}. More precisely, we define $Q_{V|Z} \in \mathcal{P}(\mathcal{Z} \times \mathcal{Z})$ as the *personalization distribution* that reflects the diversity of users in the same cluster. For each user i, his/her personalized rating $V_{ij} \in \mathcal{Z}$ to item j is distributed according to $Q_{V|Z=\mathsf{N}_{ij}} \in \mathcal{P}(\mathcal{Z})$. A natural assumption we adopt is that $Q_{V|Z=z}(z) > Q_{V|Z=z}(z')$ for all $z' \neq z$; that is, if the nominal rating is $z \in \mathcal{Z}$, then the personalized rating is *most likely* to be z. For a specific user cluster $\mathcal{U}_a$ and an item cluster $\mathcal{I}_b$, all the personalized ratings $\{V_{ij}\}_{i \in \mathcal{U}_a, j \in \mathcal{I}_b}$ follow the same distribution $Q_{V|Z=z_{ab}}$. For simplicity, we abbreviate $Q_{V|Z=z_{ab}}$ as Q_{ab}. An example of the personalized rating matrix is illustrated in Fig. 1(b).

A. Observations

The learner observes three pieces of information:

1) A sub-sampled rating matrix U, with each entry $\mathsf{U}_{ij} = V_{ij}$ with probability (w.p.) p and $\mathsf{U}_{ij} = e$ (erasure symbol) w.p. $1 - p$. We refer to p as the *sample probability* and mnp as the expected number of sampled entries.
2) A social graph $G_1 = (\mathcal{V}_1, \mathcal{E}_1)$, where $\mathcal{V}_1$ is the set of n user nodes. Let B be a $k_1 \times k_1$ symmetric *connectivity matrix* that represents the probabilities of connecting two nodes in G_1. Each pair of nodes (i, i') is connected (i.e., $(i, i') \in \mathcal{E}_1$) independently w.p. $\mathsf{B}_{\sigma(i)\sigma(i')}$.
3) An item graph $G_2 = (\mathcal{V}_2, \mathcal{E}_2)$, where $\mathcal{V}_2$ is the set of m item nodes. Let B' be a $k_2 \times k_2$ symmetric connectivity matrix that represents the probabilities of connecting two nodes in G_2. Each pair of nodes (j, j') is connected (i.e., $(j, j') \in \mathcal{E}_2$) independently w.p. $\mathsf{B}'_{\tau(j)\tau(j')}$.

Note that the leaner can only observe the nodes and edges of the graphs G_1 and G_2, without knowing which node belongs to which cluster.

[1]For any integer $s \geq 1$, let $[s]$ represent the set of integers $\{1, \ldots, s\}$.

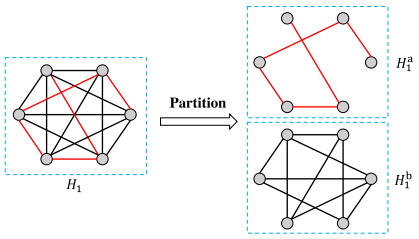


Fig. 2. The partition of a complete graph H_1 with $n = 6$ nodes into two sub-graphs H_1^{a} and H_1^{b}.

B. Objective

The learner is tasked to design an estimator $\phi = \phi(\mathsf{U}, G_1, G_2)$ to exactly recover both the user clusters $\{\mathcal{U}_a\}_{a\in[k_1]}$ and item clusters $\{\mathcal{I}_b\}_{b\in[k_2]}$ (or equivalently, the label functions σ and τ), as well as to reconstruct the nominal matrix N. The output of the estimator ϕ is denoted by $(\widehat{\sigma}, \widehat{\tau}, \widehat{\mathsf{N}})$.

To measure the accuracies of the estimated label functions $\widehat{\sigma}$ and $\widehat{\tau}$, we define the *misclassification proportions* as

$$l_1(\widehat{\sigma}, \sigma) := \min_{\pi\in\mathcal{S}_{k_1}} \frac{1}{n} \sum_{i\in[n]} \mathbb{1}\{\widehat{\sigma}(i) \neq \pi(\sigma(i))\}, \tag{1}$$

$$l_2(\widehat{\tau}, \tau) := \min_{\pi\in\mathcal{S}_{k_2}} \frac{1}{m} \sum_{j\in[m]} \mathbb{1}\{\widehat{\tau}(j) \neq \pi(\tau(j))\}, \tag{2}$$

where $\mathcal{S}_{k_1}$ (resp. $\mathcal{S}_{k_2}$) is the set of all permutations of $[k_1]$ (resp. $[k_2]$). The permutations are introduced because it is only possible to recover the *partitions* of users/items, rather than the actual labels (i.e., the best we can hope for is to ensure $l_1(\widehat{\sigma}, \sigma) = 0$ and $l_2(\widehat{\tau}, \tau) = 0$).

Furthermore, we also define the concept of *weak recovery* which plays a role in the intermediate steps of our algorithm.

Definition 1: An estimate $\widehat{\sigma}$ (resp. $\widehat{\tau}$) is said to achieve *weak recovery* if the misclassification proportion $l_1(\widehat{\sigma}, \sigma) \to 0$ as n tends to infinity (resp. $l_2(\widehat{\tau}, \tau) \to 0$ as m tends to infinity).

III. MC2G: A Computationally Efficient, Statistically Optimal Algorithm

In this section, we present a computationally efficient multi-stage algorithm called MC2G for recovering the clusters of users and items, and the nominal matrix N. Knowledge of the model parameters (e.g., connectivity matrices B and B' and personalization distribution $Q_{V|Z}$) is not needed for MC2G to succeed, as they will be estimated on-the-fly. Roughly speaking, MC2G consists of four stages: Stage 1 achieves *weak recovery* of the user/item clusters; Stage 2 estimates the model parameters B, B', and $Q_{V|Z}$; and Stages 3 and 4 respectively refine these estimates of users and items via *local refinements* steps. The inputs include the sub-sampled rating matrix U and two graphs G_1 and G_2.

Before describing our algorithm in Subsection III-B, we first point out that we use an *information splitting* method (inspired by [32], [34], [43]) to circumvent the diffculty of analyzing the error probability of our multi-stage algorithm. As a concrete example, Fig. 2 illustrates how we split the information of the social graph into two pieces, where the first piece is for Stage 1 and the second piece is for subsequent stages.

A. Information Splitting

The high-level idea is to split the observations (U, G_1, G_2) into two parts—the first part, denoted as $(G_1^{\mathrm{a}}, G_2^{\mathrm{a}})$, is used for weak recovery of users and items in Stage 1; while the second part, denoted as $(\mathsf{U}, G_1^{\mathrm{b}}, G_2^{\mathrm{b}})$, is used for estimating the parameters and for local refinements (exact recovery) of each user and item in Stages 2–4. We elaborate on the information splitting method as follows.

1) Let $H_1 = (\mathcal{V}_1, \bar{\mathcal{E}}_1)$ be the *complete graph* with vertex set $\mathcal{V}_1 = [n]$ and edge set $\bar{\mathcal{E}}_1$ which contains all the $\binom{|\mathcal{V}_1|}{2}$ edges on $\mathcal{V}_1$. We randomly partition H_1 into two sub-graphs $H_1^{\mathrm{a}} = (\mathcal{V}_1, \bar{\mathcal{E}}_1^{\mathrm{a}})$ and $H_1^{\mathrm{b}} = (\mathcal{V}_1, \bar{\mathcal{E}}_1^{\mathrm{b}})$ such that H_1^{a} is an *Erdős-Rényi (ER) graph* on $\mathcal{V}_1$ with edge probability $1/\sqrt{\log n}$. That is, each $e \in \bar{\mathcal{E}}_1$ is sampled (independently) to $\bar{\mathcal{E}}_1^{\mathrm{a}}$ with probability $1/\sqrt{\log n}$, and to $\bar{\mathcal{E}}_1^{\mathrm{b}}$ with probability $1 - 1/\sqrt{\log n}$, where $\bar{\mathcal{E}}_1^{\mathrm{b}}$ is the complement of $\bar{\mathcal{E}}_1^{\mathrm{a}}$. An example is illustrated in Fig. 2. This partition is done independently of the generation of the SBM G_1. For any realizations $H_1^{\mathrm{a}} = h_1^{\mathrm{a}}$ and $H_1^{\mathrm{b}} = h_1^{\mathrm{b}}$, let
$$G_1^{\mathrm{a}} := h_1^{\mathrm{a}} \cap G_1 \ \text{ and } \ G_1^{\mathrm{b}} := h_1^{\mathrm{b}} \cap G_1. \tag{3}$$
be two sub-SBMs on sub-graphs h_1^{a} and h_1^{b}, respectively.[2]
2) Similarly, let $H_2 = (\mathcal{V}_2, \bar{\mathcal{E}}_2)$ be the complete graph with vertex set $\mathcal{V}_2 = [m]$ and edge set $\bar{\mathcal{E}}_2$, H_2^{a} is an ER graph on $\mathcal{V}_2$ with edge probability $1/\sqrt{\log m}$, and $\bar{\mathcal{E}}_2^{\mathrm{b}}$ is the complement of $\bar{\mathcal{E}}_2^{\mathrm{a}}$. For any $H_2^{\mathrm{a}} = h_2^{\mathrm{a}}$ and $H_2^{\mathrm{b}} = h_2^{\mathrm{b}}$, we also define
$$G_2^{\mathrm{a}} := h_2^{\mathrm{a}} \cap G_2 \ \text{ and } \ G_2^{\mathrm{b}} := h_2^{\mathrm{b}} \cap G_2. \tag{4}$$

We refer the readers to Remark 7 for a discussion of the benefit of using this information splitting method.

B. Algorithm Description

Stage 1 (Weak recovery of clusters): We run a spectral clustering method[3] (e.g., Agorithm 2 in [48]) on the social graph G_1^{a} to obtain an initial estimate of the label function σ (denoted by $\sigma^{(0)}$), and also run a spectral clustering method on the item graph G_2^{a} to obtain an initial estimate of the label function τ (denoted by $\tau^{(0)}$). The estimated user clusters corresponding to $\sigma^{(0)}$ are denoted by $\{\mathcal{U}_a^{(0)}\}_{a\in[k_1]}$ (i.e., $\mathcal{U}_a^{(0)} = (\sigma^{(0)})^{-1}(a)$), and the estimated item clusters corresponding to $\tau^{(0)}$ are denoted by $\{\mathcal{I}_b^{(0)}\}_{b\in[k_2]}$. These initial estimates $\sigma^{(0)}$ and $\tau^{(0)}$ are expected to serve as good approximations of the true clusters, such that

[2]With a slight abuse of notations, we use $h_1^{\mathrm{a}} \cap G_1$ (resp. $h_1^{\mathrm{b}} \cap G_1$) to represent a graph with edge set being the intersection between the edge sets of h_1^{a} (resp., h_1^{b}) and G_1. More specifically, for the sub-SBM G_1^{a} (resp. G_1^{b}), any pairs of nodes (i, i') are connected with probability $\mathsf{B}_{\sigma_i\sigma_{i'}}$ if $(i, i') \in \bar{\mathcal{E}}_1^{\mathrm{a}}$ (resp. $(i, i') \in \bar{\mathcal{E}}_1^{\mathrm{b}}$), and with probability zero otherwise.

[3]To achieve weak recovery of the clusterings of users and items, one can also apply different variants of spectral clustering methods [34], [43], [44], semidefinite programming-based methods [45], belief propagation-based methods [46], or non-backtracking matrix-based methods [47].

Algorithm 1: MC2G.

Input : $(\mathsf{U}, G_1, G_2) = (G_1^{\mathrm{a}}, G_2^{\mathrm{a}}) \cup (\mathsf{U}, G_1^{\mathrm{b}}, G_2^{\mathrm{b}})$
Output: Clusters $\{\widehat{\mathcal{U}}_a\}_{a\in[k_1]}$ and $\{\widehat{\mathcal{I}}_b\}_{b\in[k_2]}$ (or label functions $\widehat{\sigma}$ and $\widehat{\tau}$), nominal matrix $\widehat{\mathsf{N}}$

Stage 1 (Weak recovery of communities)
Apply the spectral clustering method on G_1^{a} and G_2^{a} to obtain initial estimates $\{\mathcal{U}_a^{(0)}\}_{a\in[k_1]}$ and $\{\mathcal{I}_b^{(0)}\}_{b\in[k_2]}$;

Stage 2 (Parameters estimation)
Estimate connectivity matrices B and B' as per (5)-(6);
Estimate personalization distribution $\{\widehat{Q}_{ab}\}$ as per (7);

Stage 3 (Local refinements of users)
for *user* $i = 1$ **to** n **do**
  Calculate likelihood functions $\{L_a(i)\}_{a\in[k_1]}$;
  Let $a_i^* = \arg\max_{a\in[k_1]} L_a(i)$, and declare $i \in \mathcal{U}_{a_i^*}$;
end

Stage 4 (Local refinements of items)
for *item* $j = 1$ **to** m **do**
  Calculate likelihood functions $\{L'_b(j)\}_{b\in[k_2]}$;
  Let $b_j^* = \arg\max_{b\in[k_2]} L'_b(j)$, and declare $j \in \mathcal{I}_{b_j^*}$;
end
Reconstruct the nominal matrix $\widehat{\mathsf{N}}$ as Per (10).

both $\sigma^{(0)}$ and $\tau^{(0)}$ satisfy the weak recovery criteria defined in Definition 1.

Stage 2 (Parameters estimation): For any two sets of nodes $\mathcal{V}$ and $\mathcal{V}'$, the number of edges connecting $\mathcal{V}$ and $\mathcal{V}'$ (in G_1^{b} or G_2^{b}) is denoted as $e(\mathcal{V},\mathcal{V}')$, and the total possible number of edges connecting $\mathcal{V}$ and $\mathcal{V}'$ is denoted as $\Upsilon(\mathcal{V},\mathcal{V}')$, which equals either $|\mathcal{V}|\cdot|\mathcal{V}'|$ (if $\mathcal{V}\neq\mathcal{V}'$) or $\binom{|\mathcal{V}|}{2}$ (if $\mathcal{V}=\mathcal{V}'$). Based on the initial estimates $\{\mathcal{U}_a^{(0)}\}_{a\in[k_1]}$ and $\{\mathcal{I}_b^{(0)}\}_{b\in[k_2]}$, we then estimate the connectivity matrices B and B' as follows:

$$\widehat{\mathsf{B}}_{aa'} = e(\mathcal{U}_a^{(0)},\mathcal{U}_{a'}^{(0)})/\Upsilon(\mathcal{U}_a^{(0)},\mathcal{U}_{a'}^{(0)}), \tag{5}$$

$$\widehat{\mathsf{B}}'_{bb'} = e(\mathcal{I}_b^{(0)},\mathcal{I}_{b'}^{(0)})/\Upsilon(\mathcal{I}_b^{(0)},\mathcal{I}_{b'}^{(0)}). \tag{6}$$

Moreover, we define $\mathcal{Q}_{ab}^z := \{(i,j) : \mathsf{U}_{ij} = z, i \in \mathcal{U}_a^{(0)}, j \in \mathcal{I}_b^{(0)}\}$ for $a \in [k_1]$, $b \in [k_2]$, and $z \in \mathcal{Z}$. Then, the estimated personalization distribution is given by

$$\widehat{Q}_{ab}(z) = |\mathcal{Q}_{ab}^z| / \sum_{z\in\mathcal{Z}} |\mathcal{Q}_{ab}^z|, \quad \forall a \in [k_1], b \in [k_2]. \tag{7}$$

Stage 3 (Local refinements of users): This stage refines the classification of each user locally, based on the ratings in U, the social graph G_1^{b}, and the initial estimates $\{\mathcal{U}_a^{(0)}\}_{a\in[k_1]}$ and $\{\mathcal{I}_b^{(0)}\}_{b\in[k_2]}$. For each user $i \in [n]$, we essentially adopt a *local MLE* to determine which cluster it belongs to. We define the *likelihood function* that reflects how likely user i belongs to cluster $\mathcal{U}_a$ as:

$$L_a(i) := \sum_{a'\in[k_1]} e(\{i\},\mathcal{U}_{a'}^{(0)}) \cdot \log\left(\widehat{\mathsf{B}}_{aa'}/(1-\widehat{\mathsf{B}}_{aa'})\right) + \sum_{b\in[k_2]} \sum_{j\in\mathcal{I}_b^{(0)}} \mathbb{1}\{\mathsf{U}_{ij}\neq \mathsf{e}\} \cdot \log \widehat{Q}_{ab}(\mathsf{U}_{ij}). \tag{8}$$

Let $a_i^* := \arg\max_{a\in[k_1]} L_a(i)$ be the index of the most likely user cluster for user i. MC2G then declares $i \in \widehat{\mathcal{U}}_{a_i^*}$; or equivalently, $\widehat{\sigma}(i) = a_i^*$.

Stage 4 (Local refinements of items): This stage refines the classification of each item locally, based on U, G_2^{b}, and the initial estimates $\{\mathcal{U}_a^{(0)}\}_{a\in[k_1]}$ and $\{\mathcal{I}_b^{(0)}\}_{b\in[k_2]}$. For each item $j \in [m]$, we define the likelihood function that reflects how likely item j belongs to cluster $\mathcal{I}_b$ as:

$$L'_b(j) := \sum_{b'\in[k_2]} e(\{j\},\mathcal{I}_{b'}^{(0)}) \cdot \log\left(\widehat{\mathsf{B}}'_{bb'}/(1-\widehat{\mathsf{B}}'_{bb'})\right) + \sum_{a\in[k_1]} \sum_{i\in\mathcal{U}_a^{(0)}} \mathbb{1}\{\mathsf{U}_{ij}\neq \mathsf{e}\} \cdot \log \widehat{Q}_{ab}(\mathsf{U}_{ij}). \tag{9}$$

Let $b_j^* := \arg\max_{b\in[k_2]} L'_b(j)$ be the index of the most likely item cluster for item j. MC2G then declares $j \in \widehat{\mathcal{I}}_{b_j^*}$; or equivalently, $\widehat{\tau}(j) = b_j^*$.

Finally, one can recover the nominal matrix $\widehat{\mathsf{N}}$ by setting

$$\widehat{\mathsf{N}}_{ij} = \arg\max_{z\in\mathcal{Z}} \widehat{Q}_{ab}(z), \quad \text{for } i \in \widehat{\mathcal{U}}_a, j \in \widehat{\mathcal{I}}_b. \tag{10}$$

Remark 1: It is worth pointing out that our algorithm MC2G can be viewed as an efficient method to find an approximated solution of the maximum likelihood (ML) estimator. To be specific, the log-likelihood function of the observations (U, G_1, G_2) conditioned on the parameters $(\{\mathcal{U}_a\},\{\mathcal{I}_b\},\mathsf{B},\mathsf{B}',\{Q_{ab}\})$ is given by

$$\log \mathbb{P}(\mathsf{U}, G_1, G_2) = \log \mathbb{P}(\mathsf{U}) + \log \mathbb{P}(G_1) + \log \mathbb{P}(G_2), \tag{11}$$

$$\text{where } \log \mathbb{P}(G_1) = \sum_{a=1}^{k_1}\sum_{a'=a}^{k_1} E(\mathcal{U}_a,\mathcal{U}_{a'}) \log \mathsf{B}_{aa'} + [\Upsilon(\mathcal{U}_a,\mathcal{U}_{a'}) - E(\mathcal{U}_a,\mathcal{U}_{a'})] \log(1-\mathsf{B}_{aa'}), \tag{12}$$

$$\log \mathbb{P}(G_2) = \sum_{b=1}^{k_2}\sum_{b'=b}^{k_2} E(\mathcal{I}_b,\mathcal{I}_{b'}) \log \mathsf{B}'_{bb'} + [\Upsilon(\mathcal{I}_b,\mathcal{I}_{b'}) - E(\mathcal{I}_b,\mathcal{I}_{b'})] \log(1-\mathsf{B}'_{bb'}), \tag{13}$$

$$\log \mathbb{P}(\mathsf{U}) = \sum_{a=1}^{k_1}\sum_{b=1}^{k_2}\sum_{i\in\mathcal{U}_a}\sum_{j\in\mathcal{I}_b} \mathbb{1}\{\mathsf{U}_{ij} = \mathsf{e}\} \log(1-p) + \mathbb{1}\{\mathsf{U}_{ij} \neq \mathsf{e}\} \log\left(pQ_{ab}(\mathsf{U}_{ij})\right). \tag{14}$$

Here, $E(\mathcal{U}_a,\mathcal{U}_{a'})$ and $E(\mathcal{I}_b,\mathcal{I}_{b'})$ respectively denote the number of edges connecting the two clusters in graphs G_1 and G_2, which are close to $e(\mathcal{U}_a,\mathcal{U}_{a'})$ and $e(\mathcal{I}_b,\mathcal{I}_{b'})$ since the sub-graphs G_1^{b} and G_2^{b} are almost as dense as the original graphs G_1 and G_2. Based on the log-likelihood functions in (11)–(14), one can check that each of the Stages 2–4 approximately optimizes $\log \mathbb{P}(\mathsf{U}, G_1, G_2)$ over a specific subset of parameters while keeping other parameters fixed. This can be viewed as an intuitive interpretation of Stages 2–4.

Remark 2: To improve the performance of MC2G in practical scenarios, one can perform parameters estimation (Stage 2) and local refinements (Stages 3–4) iteratively for multiple rounds. After one round of local refinements, the parameters will be estimated more accurately, which will in turn benefit the next round of local refinements. Accordingly, the computational complexity will increase.

From a theoretical perspective, when applying MC2G to the symmetric setting described in Section IV, it suffices to perform parameters estimation and local refinements *only once*. This is because (i) Lemmas 2, 3, and 5 together show that the estimates of the connectivity matrices and the personalization distributions are accurate enough such that in the following local refinement stages, the estimation errors are negligible; and (ii) Lemmas 4 and 6 show that one round of local refinements is sufficient for the exact recovery of the clusters of users and items with high probability.

Remark 3: The information splitting method is merely for the purpose of analysis, while it may not be practical when n and m are *not sufficiently large*, in which case the first part of the graphs $(G_1^{\mathrm{a}}, G_2^{\mathrm{a}})$ may be too sparse to achieve weak recovery of the true clusters in Stage 1. Thus, in practice, one can skip the information splitting step in Section III-A and simply apply every stage on the fully-observed graphs (G_1, G_2) for weak recovery, parameter estimations, and local refinements—this is referred to as the *simplified version of* MC2G. In our experiments (Section VII), we use this simplified version of MC2G and show that it also works well empirically.

C. Computational Complexity

Using the iterative power method [49], the spectral clustering method in Stage 1 runs in times at most $\mathcal{O}(|\mathcal{E}_1|\log n)$ and $\mathcal{O}(|\mathcal{E}_2|\log m)$ respectively. In each of the following steps, MC2G requires (at most) a *single* pass of all the sub-sampled entries in the rating matrix U and the edge sets $\mathcal{E}_1$ and $\mathcal{E}_2$, which amounts to at most $\mathcal{O}(\max\{|\mathcal{E}_1|, |\mathcal{E}_2|, mnp\})$ time, where p is the sample probability. Thus, the overall computational complexity is $\mathcal{O}(\max\{|\mathcal{E}_1|\log n, |\mathcal{E}_2|\log m, mnp\})$. Finally, we would like to point out that for most scenarios of interest (including the following theoretical setting and experimental setting), the numbers of edges in the social and item graphs satisfy $|\mathcal{E}_1| = \mathcal{O}(n\log n)$ and $|\mathcal{E}_2| = \mathcal{O}(m\log m)$ with high probability, and the sample complexity $mnp = \mathcal{O}(\max\{n\log n, m\log m\})$. Therefore, the overall complexity is quasilinear in n and m with high probability.

IV. Theoretical Guarantees of MC2G and Information-Theoretic Lower Bounds

This section provides theoretical guarantees for MC2G. Under the symmetric setting defined in Subsection IV-A, we characterize the expected number of sampled entries required for MC2G to succeed; the key message there is that this quantity depends critically on (i) the "qualities" of the social and item similarity graphs, and (ii) the *squared Hellinger distance* between the rating statistics of different user/item clusters. We further establish an information-theoretic lower bound on the expected number of sampled entries. This bound matches the achievability bound up to a constant factor, thus demonstrates the order-wise optimality of MC2G.

A. The Symmetric Setting

Under the symmetric setting, it is assumed that (i) the user clusters are of equal size (i.e., $|\mathcal{U}_a| = n/k_1$ for all $a \in [k_1]$) and the item clusters are of equal size (i.e., $|\mathcal{I}_b| = m/k_2$ for all $b \in [k_2]$),[4] and (ii) the connection probability for each pair of nodes depends only on whether they belong to the same cluster, i.e., the connectivity matrices B and B' satisfy

$$\mathsf{B}_{aa'} = \begin{cases} \alpha_1, & \text{if } a = a'; \\ \beta_1, & \text{if } a \neq a'; \end{cases} \text{ and } \mathsf{B}'_{bb'} = \begin{cases} \alpha_2, & \text{if } b = b'; \\ \beta_2, & \text{if } b \neq b'. \end{cases}$$

Following many prior works on the SBM [32], [34], [38], [39], we consider the *logarithm average degree regime* in which α_1 and β_1 scale as $\Theta((\log n)/n)$ such that each node has expected degree $\Theta(\log n)$. This regime is of particular interest because (i) it is known that the threshold for exact recovery of clusters falls into this regime [32], [34]; and (ii) as we shall see in Theorems 1 and 2, the gain of G_1 can be precisely characterized in this regime. Similarly, we also assume that α_2 and β_2 scale as $\Theta((\log m)/m)$. Moreover, following the prior work [12], we assume $m = \omega(\log n)$ and $n = \omega(\log m)$ such that $m \to \infty$ as $n \to \infty$.

We note that MC2G is not restricted to the symmetric setting; it can be applied more generally to asymmetric scenarios. Indeed, for the experiments in Section VII, we do not make the symmetric assumption. In this section, however, we make this assumption to simplify the presentation of Theorem 1 and to clearly understand the effect of the parameters of the model on the minimum expected number of sampled entries required for MC2G to succeed.

In the following, we formally define the notion of *exact recovery*. Note that the model is governed by the pair of label functions (σ, τ) together with the nominal matrix N, and we define the *parameter space* that contains all valid $(\sigma, \tau, \mathsf{N})$ under the symmetric setting as

$$\begin{aligned}\Xi \triangleq \Big\{(\sigma,\tau,\mathsf{N}) \big| & \sigma:[n]\to[k_1],\ \big|\{i\in[n]:\sigma_i=a\}\big| = \frac{n}{k_1}, \forall a\in[k_1]; \\ & \tau:[m]\to[k_2],\ \big|\{j\in[m]:\tau_j=b\}\big| = \frac{m}{k_2},\ \forall b\in[k_2]; \\ & \mathsf{N}\in\mathcal{Z}^{n\times m}, \mathsf{N}_{ij} = \mathsf{N}_{i'j'} \text{ if } \sigma(i)=\sigma(i') \text{ and } \tau(j)=\tau(j')\Big\}.\end{aligned}$$

Let $(\sigma, \tau, \mathsf{N})$ be the ground truth, and $(\widehat{\sigma}, \widehat{\tau}, \widehat{\mathsf{N}})$ be the output of the estimator $\phi = \phi(\mathsf{U}, G_1, G_2)$. We say the event $\mathcal{E}_{(\sigma,\tau,\mathsf{N})}$ occurs if the output $(\widehat{\sigma}, \widehat{\tau}, \widehat{\mathsf{N}})$ of the estimator ϕ satisfies one of the following three criterions: (i) $\{l_1(\widehat{\sigma}, \sigma) \neq 0\}$, (ii) $\{l_2(\widehat{\tau}, \tau) \neq 0\}$, and (iii) $\{\widehat{\mathsf{N}} \neq \mathsf{N}\}$.

[4] We implicitly assume that n is divisible by k_1 and m is divisible by k_2. In the case that n and m are not multiples of k_1 and k_2 respectively, rounding operations required to define the set Ξ. Such rounding operations, however, do not affect the calculations and results downstream.

Definition 2 (Exact recovery): For any estimator ϕ, its corresponding (maximum) error probability is defined as

$$P_{\text{err}}(\phi) := \max_{(\sigma,\tau,\mathsf{N})\in\Xi} \mathbb{P}_{(\sigma,\tau,\mathsf{N})}\left(\phi(\mathsf{U}, G_1, G_2) \in \mathcal{E}_{(\sigma,\tau,\mathsf{N})}\right),$$

where $\mathbb{P}_{(\sigma,\tau,\mathsf{N})}(\cdot)$ is the probability when (U, G_1, G_2) is generated according to the model governed by $(\sigma, \tau, \mathsf{N})$. A sequence of estimators $\Phi = \{\phi_n\}_{n=1}^{\infty}$ achieves exact recovery if

$$\lim_{n\to\infty} P_{\text{err}}(\phi_n) = 0. \tag{15}$$

Definition 3 (Sample complexity): The sample complexity is defined as the minimum expected number of samples in the matrix U such that there exists Φ for which (15) holds.

B. Theoretical Guarantees of MC2G

As we shall see, the "qualities" of the social and item graphs play a key role in the performance of MC2G. Specifically, we define a measure of the quality of the social graph G_1 as $I_1 := n(\sqrt{\alpha_1} - \sqrt{\beta_1})^2/(\log n)$. A larger value of I_1 implies a better quality of the graph, since the structures of the clusters are more clearly delineated when the difference between the intra-cluster probability α_1 and the inter-cluster probability β_1 is larger. Analogously, we define a measure of the quality of the item graph G_2 as $I_2 := m(\sqrt{\alpha_2} - \sqrt{\beta_2})^2/(\log m)$.

The performance of MC2G also depends on the statistics of the rating matrix. Intuitively, if the rating statistics of two clusters are further apart, it is then easier to distinguish them. It turns out that under the symmetric setting, the distance between the rating statistics of different clusters can be measured by the *squared Hellinger distance:* $H^2(P,Q) := 1 - \sum_{z\in\mathcal{Z}} \sqrt{P(z)Q(z)}$, for probability distributions P and Q. We then define $d(\mathcal{U}_a, \mathcal{U}_{a'}) := \sum_{b\in[k_2]} H^2(Q_{ab}, Q_{a'b})$ as a measure of the discrepancy between user clusters $\mathcal{U}_a$ and $\mathcal{U}_{a'}$ (where $a, a' \in [k_1]$), and $d_{\mathcal{U}} := \min_{a\neq a'} d(\mathcal{U}_a, \mathcal{U}_{a'})$ as the minimal discrepancy over all pairs of user clusters. A larger value of $d_{\mathcal{U}}$ means that it is easier to distinguish all the user clusters. Analogously, we define the discrepancy between item clusters $\mathcal{I}_b$ and $\mathcal{I}_{b'}$ (where $b, b' \in [k_2]$) as $d(\mathcal{I}_b, \mathcal{I}_{b'}) := \sum_{a\in[k_1]} H^2(Q_{ab}, Q_{ab'})$, and $d_{\mathcal{I}} := \min_{b\neq b'} d(\mathcal{I}_b, \mathcal{I}_{b'})$ as the minimal discrepancy over all pairs of item clusters.

Remark 4: The squared Hellinger distance $H^2(P,Q)$ satisfies $H^2(P,Q) \in [0,1]$ and $H^2(P,Q) = 0$ if and only if $P = Q$.

Theorem 1 states the expected number of sampled entries needed for MC2G to succeed under the symmetric setting.

Theorem 1 (Performance of MC2G): For any $\epsilon > 0$, if the expected number of sampled entries mnp satisfies

$$mnp \geq \max\left\{\frac{\left\lfloor(1+\epsilon) - \frac{I_1}{k_1}\right\rfloor n\log n}{d_{\mathcal{U}}/k_2}, \frac{\left\lfloor(1+\epsilon) - \frac{I_2}{k_2}\right\rfloor m\log m}{d_{\mathcal{I}}/k_1}\right\}, \tag{16}$$

then MC2G ensures $P_{err} \to 0$ as $n \to \infty$.

Remark 5: Some remarks on Theorem 1 are in order.

1) Roughly speaking, the first term on the RHS of (16) is the threshold for Stage 3 (local refinements of users) to succeed. This is because when mnp exceeds the first term, the probability that a single user is misclassified to an incorrect cluster (in Stage 3) is at most $n^{-\ell}$ for some $\ell > 1$. Thus, taking a union bound over all the n users still results in a vanishing error probability. Similarly, the second term on the RHS of (16) is the threshold for Stage 4 to succeed.
2) Our result in (16) confirms our intuitive belief that increasing $d_{\mathcal{U}}$ and $d_{\mathcal{I}}$ (the minimum discrepancies between user and item clusters) indeed helps to reduce the number of samples required for exact recovery. Similarly, increasing I_1 and I_2 (the qualities of the social and item graphs) also helps to reduce the sample complexity.
3) It is also worth noting that when $I_1 > k_1$, the first term in (16) becomes non-positive (thus inactive); this means that performing local refinements of users in Stage 3 is no longer needed, which is due to the fact that Stage 1 has already ensured exact recovery of k_1 user clusters. This observation coincides with the theoretical result of community detection in the symmetric SBM [34], which states that exact recovery of k_1 clusters is possible when $I_1 > k_1$. Similarly, when $I_2 > k_2$, local refinements of items in Stage 4 is no longer needed, as Stage 1 has ensured exact recovery of k_2 item clusters.
4) While the theoretical result in Theorem 1 is dedicated to this symmetric setting, MC2G is applicable to a more general matrix completion problem with social and item similarity graphs, where the sizes of user/item clusters may be different. This is confirmed by the experiments in Section VII.

C. Information-Theoretic Lower Bound

Theorem 2 below provides an information-theoretic lower bound on the sample complexity under the symmetric setting. Again, the lower bound is a function of I_1, I_2 (the quality of the social/item graph), and $d_{\mathcal{U}}$ and $d_{\mathcal{I}}$ (the minimum discrepancies measured in terms of the squared Hellinger distances of user/item clusters).

Theorem 2 (Impossibility result): For any $\epsilon > 0$, if

$$mnp < \max\left\{\frac{\left\lfloor\frac{1-\epsilon}{2} - \frac{I_1}{k_1}\right\rfloor n\log n}{d_{\mathcal{U}}/k_2}, \frac{\left\lfloor\frac{1-\epsilon}{2} - \frac{I_2}{k_2}\right\rfloor m\log m}{d_{\mathcal{I}}/k_1}\right\}, \tag{17}$$

then $\lim_{n\to\infty} P_{err}(\phi) = 1$ for any estimator ϕ.

Theorem 2 states that *any* estimator *must* necessarily fail if the expected number of samples is smaller than the maximal term in (17). Thus, the sample complexity defined in Definition 3 is upper-bounded by the RHS of (16), and lower-bounded by the RHS of (17). In particular, Theorem 2 guarantees that P_{err} approaches one as $n \to \infty$; this is the so-called *strong converse* in the information theory parlance. Comparing (17) with the achievability bound in (16), we note that they match up to a constant factor, and this further demonstrates the order-wise optimality of the proposed computationally efficient algorithm MC2G.

D. Comparisons With Other Results on Matrix Completion

The model considered in this work can be viewed as a special *noisy matrix completion* [50]–[52] problem, where the nominal matrix N is not only of low rank, but also has a block structure imposed by the clusters of rows and columns. Here, the nominal matrix N is analogous to the low-rank matrix to be recovered, the personalization distributions $\{Q_{ab}\}$ is analogous to the noise, and the observed rating matrix U is analogous to the observed perturbed matrix.

The method of *weighted nuclear norm and ℓ_1 norm minimization* [50] can be directly applied to recover the nominal matrix N; in order to exactly recover N with high probability, the number of sampled entries should be at least $\Omega(\max\{n\log^6 n, m\log^6 m\})$ [50, Theorem 1]. In contrast, MC2G only needs to sample $\Theta(\max\{n\log n, m\log m\})$ entries (Theorem 1), which is better than their method by a factor of $\log^5 n$ or $\log^5 m$. Moreover, Theorem 1 also shows that the constant in $\Theta(\max\{n\log n, m\log m\})$ can further be reduced by exploiting the available graphs.

We note that other works [15], [51]–[53] also studied matrix completion in the presence of noise, and provided algorithms that are applicable to our problem. However, their theoretical results focus on bounding the estimation error between the true matrix N and recovered matrix $\widehat{\mathsf{N}}$, thus they are not directly comparable to our result.

In the absence of noise, theoretical guarantees of exact recovery for various matrix completion algorithms have been well understood. Assume the rank of the matrix is $\mathcal{O}(1)$. For nuclear norm minimization-based methods [19], [54], [55], it has been shown that $\mathcal{O}(\max\{n\log^2 n, m\log^2 m\})$ sampled entries are sufficient for exact recovery. For gradient descent-based methods such as OPTSPACE, $\mathcal{O}(\max\{n\log n, m\log m\})$ sampled entries suffice [15].

Unlike the aforementioned algorithms, there are also algorithms exploiting both matrix entries and graph side information. For example, Ahn *et al.* [1] developed a two-stage algorithm that incorporates a social graph, and provided theoretical guarantees for a simple setting with binary matrices and two user clusters. For a fair comparison, we consider the following setting: $k_1 = k_2 = 2$ (two user/item clusters), nominal ratings $z_{11} = z_{22} = 1$, $z_{12} = z_{21} = 0$, $\mathcal{Z} = \{0, 1\}$, and the personalization distribution $Q_{V|Z}(v|z)$ equals $1-\theta$ if $v = z$, and θ otherwise. To exactly recover the matrix, their algorithm requires $g(\theta)\cdot\max\{(1-\frac{I_1}{2})n\log n, 2m\log m\}$ sampled entries [1, Theorem 2], while MC2G requires $g(\theta)\cdot\max\{(1-\frac{I_1}{2})n\log n, (1-\frac{I_2}{2})m\log m\}$ sampled entries, where $g(\theta) = (1-2\sqrt{\theta(1-\theta)})^{-1}$. When m is relatively large, we note that MC2G reduces at least by $g(\theta)\cdot(1+\frac{I_2}{2})m\log m$ compared to [1], and the advantage becomes larger when the quality of the item graph (represented by I_2) becomes better. As mentioned in Section I-B, the work by Rao *et al.* [8] further considered exploiting both the social and item graphs, and they presented a novel algorithm as well as an accompanying theoretical guarantee on the estimation error $\|\widehat{\mathsf{N}} - \mathsf{N}\|_{\mathrm{F}}$. However, their problem formulation and the form of results are not directly comparable to ours, since they do not aim to exactly recover N. Nevertheless, we point out that our advantage is that we explicitly quantify the gains (by characterizing the exact constant) due to exploiting social and item similarity graphs, and we also demonstrate its order-optimality, while in [8] the benefits of exploiting the social and item graphs are not quantitatively characterized.

V. PROOF OF THEOREM 1

A. Analysis of Stage 1

Note that the sub-SBM G_1^{a} is generated on the sub-graph h_1^{a}; thus the performance of the spectral clustering method on G_1^{a} essentially depends on the realization h_1^{a}. A similar argument also applies to G_2^{a}.

To circumvent the difficulties of analyzing fixed h_1^{a} and h_1^{b}, we first consider two artificial SBMs $\widetilde{G}_1$ and $\widetilde{G}_2$, where $\widetilde{G}_1$ is generated on the n user nodes and has connectivity matrix $\mathsf{B}/\sqrt{\log n}$, and $\widetilde{G}_2$ is generated on the m item nodes and has connectivity matrix $\mathsf{B}'/\sqrt{\log m}$. In Appendix A, we introduce a result (Theorem 3) that provides theoretical guarantees of weak recovery of clusters in the SBM, which is adapted from [48, Theorem 6]. One can also check that the two artificial SBMs $\widetilde{G}_1$ and $\widetilde{G}_2$ satisfy the conditions of applying Theorem 3, as shown in Appendix VII-C. Thus, applying Theorem 3 yields that there exist vanishing sequences ϵ_n, η_n, and γ_n (depending on B and B') such that with probability at least $1-\epsilon_n$, the spectral clustering method running on $\widetilde{G}_1$ and $\widetilde{G}_2$ respectively ensure that

$$l_1(\sigma^{(0)}, \sigma) \le \eta_n \ \text{ and } \ l_2(\tau^{(0)}, \tau) \le \gamma_n. \tag{18}$$

Based on the good performances of spectral clustering methods running on $\widetilde{G}_1$ and $\widetilde{G}_2$, we next show that spectral clustering methods running on G_1^{a} and G_2^{a} also provide satisfactory initialization results with high probability.

Definition 4: Let $\mathbf{h} = (h_1^{\mathrm{a}}, h_1^{\mathrm{b}}, h_2^{\mathrm{a}}, h_2^{\mathrm{b}})$ be an aggregation of realizations of the sub-graphs.

1) A sub-graph h_1^{a} is said to be good if the probability that "a spectral clustering method running on G_1^{a} (which depends on h_1^{a}) ensures $l_1(\sigma^{(0)}, \sigma) \le \eta_n$" is at least $1-\sqrt{\epsilon_n}$. A sub-graph h_1^{b} is said to be good if the degree of any node in h_1^{b} is at least $n(1-2/\sqrt{\log n})$.
2) A sub-graph h_2^{a} is said to be good if the probability that "a spectral clustering method running on G_2^{a} ensures $l_2(\tau^{(0)}, \tau) \le \gamma_n$" is at least $1-\sqrt{\epsilon_n}$. A sub-graph h_2^{b} is said to be good if the degree of any node in h_2^{b} is at least $m(1-2/\sqrt{\log m})$.
3) Let $\mathcal{G}$ and $\mathcal{B}$ be two disjoint sets of $\mathbf{h}$. We say $\mathbf{h} \in \mathcal{G}$ if all the elements in $\mathbf{h}$ are good, and $\mathbf{h} \in \mathcal{B}$ otherwise.

Lemma 1: The randomly generated sub-graphs $H_1^{\mathrm{a}}, H_1^{\mathrm{b}}, H_2^{\mathrm{a}}, H_2^{\mathrm{b}}$ are all good with probability at least $(1-2\sqrt{\epsilon_n})^2$, i.e., $\sum_{\mathbf{h}\in\mathcal{G}} \mathbb{P}(\mathbf{h}) \ge (1-2\sqrt{\epsilon_n})^2$.

Proof: See Appendix B. □

We define $\mathcal{G}'$ as the set of label functions that are close to the true label functions (σ, τ), i.e.,

$$\mathcal{G}' := \{(\sigma,'\tau') : l_1(\sigma,'\sigma) \le \eta_n,\ l_2(\tau,'\tau) \le \gamma_n\}, \tag{19}$$

and $\mathcal{B}' := \{(\sigma,'\sigma) : (\sigma,'\sigma) \notin \mathcal{G}'\}$ as the complement of $\mathcal{G}'$. By definition, we know that when the randomly generated sub-graphs $\mathbf{h} \in \mathcal{G}$, running spectral clustering methods on G_1^{a} and

G_2^{a} yields $(\sigma^{(0)},\tau^{(0)}) \in \mathcal{G}'$ with high probability, i.e.,

$$\sum_{(\sigma^{(0)},\tau^{(0)})\in\mathcal{G}'} \mathbb{P}((\sigma^{(0)},\tau^{(0)})|\mathbf{h}) \geq (1-\sqrt{\epsilon_n})^2, \qquad (20)$$

which is uniform in $\mathbf{h} \in \mathcal{G}$ (i.e., $\{\epsilon_n\}$ does not depend on $\mathbf{h}$).

Remark 6: Lemma 1 above conveys two important messages: (i) Although the sub-graphs H_1^{a} and H_2^{a} are much sparser compared to H_1 and H_2 (or equivalently, the information contained in H_1^{a} and H_2^{a} is much less), they still guarantee the success of running spectral clustering methods (with high probability). (ii) The densities of sub-graphs H_1^{b} and H_2^{b} are almost the same as those of H_1 and H_2, and this is critical in Stages 2–4 for proving the theoretical guarantees of Mc2g.

Remark 7: After the splitting, all the random variables (associated with the original SBMs G_1 and G_2) are partitioned into two *disjoint* sets—the first set of random variables is associated with the sub-graphs h_1^{a} and h_2^{a}, and is used in Stage 1, while Stages 2–4 rely on the second set of random variables that are associated with the sub-graph h_1^{b} and h_2^{b}. The two sets of random variables are independent since each edge in the SBM is generated independently. Note that the output from Stage 1 (i.e., the initial estimates of the user and item clusters) is only a function of the first set of random variables. After obtaining the initial estimates from Stage 1, we then implement the subsequent stages on h_1^{b} and h_2^{b}. Since h_1^{b} and h_2^{b} only involve the second set of random variables, the analysis of the subsequent stages relies only on the second set of random variables, which are completely independent of the first set of random variables (and thus independent of the output from Stage 1).

In contrast, if there were no splitting process, the output from Stage 1 would be a function of *all* the random variables associated with the SBMs G_1 and G_2, while the analysis of Stage 2–4 also relies on these random variables, whose distributions could be changed conditioned on the realization of the output from Stage 1.

B. Analysis of Stage 2

Note that the estimates $\widehat{\mathsf{B}}, \widehat{\mathsf{B}}', \{\widehat{Q}_{ab}\}$ in (5)–(7) depend on both $\mathbf{h}$ and $(\sigma^{(0)},\tau^{(0)})$. In Stage 2, we show in Lemmas 2 and 3 below that conditioned on $\mathbf{h} \in \mathcal{G}$ and $(\sigma^{(0)},\tau^{(0)}) \in \mathcal{G}'$, the estimates are accurate with high probability.

Lemma 2: Suppose $\mathbf{h} \in \mathcal{G}$ and $(\sigma^{(0)},\tau^{(0)}) \in \mathcal{G}'$. With probability $1-o(1)$, there exists a sequence $\varepsilon_n \in \Omega(\max\{\gamma_n,\eta_n,1/\sqrt{\log n}\}) \cap o(1)$ such that for all $a,a' \in [k_1]$ and $b,b' \in [k_2]$, $\left|\frac{\widehat{\mathsf{B}}_{aa'}-\mathsf{B}_{aa'}}{\mathsf{B}_{aa'}}\right| \leq \varepsilon_n$ and $\left|\frac{\widehat{\mathsf{B}}'_{bb'}-\mathsf{B}'_{bb'}}{\mathsf{B}'_{bb}}\right| \leq \varepsilon_n$.

Proof: See Appendix C. □

Lemma 3: Suppose $\mathbf{h} \in \mathcal{G}$ and $(\sigma^{(0)},\tau^{(0)}) \in \mathcal{G}'$. With probability $1-o(1)$, there exists a sequence $\varepsilon'_n \in \Omega(\max\{\gamma_n,\eta_n,1/\sqrt{\log n}\}) \cap o(1)$ such that for all $a \in [k_1]$, $b \in [k_2]$, and $z \in \mathcal{Z}$, $|(\widehat{Q}_{ab}(z)/Q_{ab}(z))-1| \leq \varepsilon'_n$.

Proof: See Appendix D. □

Remark 8: In Lemmas 2 and 3 above, we implicitly assume (without loss of generality) that the permutations minimizing $l_1(\sigma^{(0)},\sigma)$ and $l_2(\tau^{(0)},\tau)$ are both the *identity permutation*, i.e., $l_1(\sigma^{(0)},\sigma) = \sum_{i\in[n]} \mathbb{1}\{\sigma^{(0)}(i) \neq \sigma(i)\}/n$ and $l_2(\tau^{(0)},\tau) = \sum_{j\in[m]} \mathbb{1}\{\tau^{(0)}(j) \neq \tau(j)\}/m$ as per (1) and (2). Without this assumption, one needs to introduce the permutations π_1^* and π_2^* that respectively minimize $l_1(\sigma^{(0)},\sigma)$ and $l_2(\tau^{(0)},\tau)$—this unnecessarily complicates the presentations of Lemmas 2 and 3.

C. Analysis of Stage 3

Note that the likelihood function defined in (8) depends on the estimated values $\widehat{\mathsf{B}}$ and $\{\widehat{Q}_{ab}\}$ of the model parameters. For ease of analysis, we first ignore the imprecisions of these estimates, and define the *exact* likelihood function $\widetilde{L}_a(i)$, which depends on the exact values of B and $\{Q_{ab}\}$, as

$$\widetilde{L}_a(i) := \sum_{a'\in[k_1]} e(\{i\},\mathcal{U}_{a'}^{(0)}) \cdot \log\left(\mathsf{B}_{aa'}/(1-\mathsf{B}_{aa'})\right) + \sum_{b\in[k_2]} \sum_{j\in\mathcal{I}_b^{(0)}} \mathbb{1}\{\mathsf{U}_{ij}\neq\mathsf{e}\} \cdot \log Q_{ab}(\mathsf{U}_{ij}). \qquad (21)$$

We now consider a specific user $i \in [n]$, which belongs to cluster $\mathcal{U}_a$ for some $a \in [k_1]$. Lemma 4 below shows that, with probability $1-o(1/n)$, $\widetilde{L}_a(i)$ is larger than any other likelihood functions $\widetilde{L}_{\bar{a}}(i)$ by at least $(\epsilon/2)\log n$.

Lemma 4: Suppose $\mathbf{h} \in \mathcal{G}$ and $(\sigma^{(0)},\tau^{(0)}) \in \mathcal{G}'$. If

$$mnp \geq \frac{[(1+\epsilon)-(I_1/k_1)]\,n\log n}{d_{\mathcal{U}}/k_2}, \qquad (22)$$

with probability at least $1-k_1 n^{-(1+\frac{\epsilon}{2})}$,

$$\widetilde{L}_a(i) > \max_{\bar{a}\in[k_1]\setminus\{a\}} \widetilde{L}_{\bar{a}}(i) + (\epsilon/2)\log n. \qquad (23)$$

Proof: In the following, it suffices to focus on the boundary case $mnp = \frac{[(1+\epsilon)-(I_1/k_1)]n\log n}{d_{\mathcal{U}}/k_2}$, since the probability that (23) holds would only be larger as the sample complexity mnp increases. Consider a specific $\bar{a} \neq a$. Under the symmetric setting, the entries in the connectivity matrix B are either α_1 or β_1, and we further define $\lambda_1 := \log\frac{(1-\beta_1)\alpha_1}{(1-\alpha_1)\beta_1}$. From the definitions of $\widetilde{L}_a(i)$ and $\widetilde{L}_{\bar{a}}(i)$ in (21), we have

$$\widetilde{L}_a(i) - \widetilde{L}_{\bar{a}}(i) = \sum_{a'\in[k_1]} e(\{i\},\mathcal{U}_{a'}^{(0)}) \cdot \log\frac{\mathsf{B}_{aa'}(1-\mathsf{B}_{\bar{a}a'})}{\mathsf{B}_{\bar{a}a'}(1-\mathsf{B}_{aa'})} + \sum_{b\in[k_2]} \sum_{j\in\mathcal{I}_b^{(0)}} \mathbb{1}\{\mathsf{U}_{ij}\neq\mathsf{e}\} \cdot \log\frac{Q_{ab}(\mathsf{U}_{ij})}{Q_{\bar{a}b}(\mathsf{U}_{ij})},$$

where

$$\log\frac{\mathsf{B}_{aa'}(1-\mathsf{B}_{\bar{a}a'})}{\mathsf{B}_{\bar{a}a'}(1-\mathsf{B}_{aa'})} = \begin{cases} \log\frac{\alpha_1(1-\beta_1)}{\beta_1(1-\alpha_1)} = \lambda_1, & \text{if } a'=a; \\ \log\frac{\beta_1(1-\alpha_1)}{\alpha_1(1-\beta_1)} = -\lambda_1, & \text{if } a'=\bar{a}; \\ \log\frac{\beta_1(1-\beta_1)}{\beta_1(1-\beta_1)} = 0, & \text{if } a'\neq\{a,a'\}. \end{cases}$$

Thus, we have

$$\widetilde{L}_a(i) - \widetilde{L}_{\bar{a}}(i) = \lambda_1 e(\{i\},\mathcal{U}_a^{(0)}) - \lambda_1 e(\{i\},\mathcal{U}_{\bar{a}}^{(0)}) + \sum_{b\in[k_2]} \sum_{j\in\mathcal{I}_b^{(0)}} \mathbb{1}\{\mathsf{U}_{ij}\neq\mathsf{e}\} \log\frac{Q_{ab}(\mathsf{U}_{ij})}{Q_{\bar{a}b}(\mathsf{U}_{ij})}. \qquad (24)$$

For $a, \bar{a} \in [k_1]$, let $\mathcal{S}_{a\bar{a}} := \mathcal{U}_a \cap \mathcal{U}_{\bar{a}}^{(0)}$ be the set of users that belong to cluster $\mathcal{U}_a$ and are classified to $\mathcal{U}_{\bar{a}}^{(0)}$ after Stage 1. By introducing random variables $\{X_k\}_{k=1}^n \overset{\text{i.i.d.}}{\sim} \text{Bern}(\alpha_1)$ and $\{Y_k\}_{k=1}^n \overset{\text{i.i.d.}}{\sim} \text{Bern}(\beta_1)$, one can rewrite $e(\{i\}, \mathcal{U}_a^{(0)}) - e(\{i\}, \mathcal{U}_{\bar{a}}^{(0)})$ as

$$\sum_{k \in \mathcal{S}_{aa}} X_k + \sum_{k \in \mathcal{U}_a^{(0)} \setminus \mathcal{U}_a} Y_k - \sum_{k \in \mathcal{U}_{\bar{a}}^{(0)} \setminus \mathcal{U}_a} Y_k - \sum_{k \in \mathcal{S}_{a\bar{a}}} X_k.$$

For $b, \bar{b} \in [k_2]$, let $\mathcal{T}_{b\bar{b}} := \mathcal{I}_b \cap \mathcal{I}_{\bar{b}}^{(0)}$ be the set of items that belong to cluster $\mathcal{I}_b$ and are classified to $\mathcal{I}_{\bar{b}}^{(0)}$ after Stage 1. By introducing random variables $\{T_{ij}\} \overset{\text{i.i.d.}}{\sim} \text{Bern}(p)$ and $\{Z_{ij}^{ab}\} \overset{\text{i.i.d.}}{\sim} Q_{ab}$, one can rewrite the second part in (24) as

$$\sum_{b \in [k_2]} \left[\sum_{j \in \mathcal{T}_{bb}} \underbrace{T_{ij} \log \frac{Q_{ab}(Z_{ij}^{ab})}{Q_{\bar{a}b}(Z_{ij}^{ab})}}_{:= A_{ij}^{ab}} + \sum_{\bar{b} \neq b} \sum_{j \in \mathcal{T}_{\bar{b}b}} \underbrace{T_{ij} \log \frac{Q_{ab}(Z_{ij}^{a\bar{b}})}{Q_{\bar{a}b}(Z_{ij}^{a\bar{b}})}}_{:= \bar{A}_{ij}^{ab,\bar{b}}} \right].$$

We then bound $\mathbb{P}(\widetilde{L}_a(i) - \widetilde{L}_{\bar{a}}(i) \leq (\epsilon/2)\log n)$ from above in (25)–(28) shown at the bottom of this page, where (26) follows from the Chernoff bound $\mathbb{P}(X \geq \kappa) \leq \min_{t \geq 0} e^{-t\kappa} \cdot \mathbb{E}(e^{tX})$ with $t = 1/2$, and (27) is due to the independence of the random variables. We now consider the exponent in (28). Since the initial estimate $(\sigma^{(0)}, \gamma^{(0)}) \in \mathcal{G}'$, by the definition of $\mathcal{G}'$ we know that the misclassification proportions satisfy $l_1(\sigma^{(0)}, \sigma) \leq \eta_n$ and $l_2(\tau^{(0)}, \tau) \leq \gamma_n$. As a result, we have

$$\frac{n}{k_1} - \eta_n n \leq |\mathcal{U}_{\bar{a}}^{(0)} \setminus \mathcal{U}_a| \leq \frac{n}{k_1}, \ \frac{n}{k_1} - \eta_n n \leq |\mathcal{S}_{aa}| \leq \frac{n}{k_1}, \quad (29)$$

$$|\mathcal{S}_{a\bar{a}}| \leq \eta_n n \text{ for } \bar{a} \neq a, \qquad |\mathcal{U}_a^{(0)} \setminus \mathcal{U}_a| \leq \eta_n n, \quad (30)$$

$$\frac{m}{k_2} - \gamma_n m \leq |\mathcal{T}_{bb}| \leq \frac{m}{k_2}, \qquad |\mathcal{T}_{\bar{b}b}| \leq \gamma_n m \text{ for } \bar{b} \neq b. \quad (31)$$

Next, we note that $\log \mathbb{E}(e^{\frac{1}{2}\lambda_1 Y_k}) = \log(1 - \beta_1 + \beta_1 \sqrt{\frac{(1-\beta_1)\alpha_1}{(1-\alpha_1)\beta_1}})$ and $\log \mathbb{E}(e^{-\frac{1}{2}\lambda_1 X_k}) = \log(1 - \alpha_1 + \alpha_1 \sqrt{\frac{(1-\alpha_1)\beta_1}{(1-\beta_1)\alpha_1}})$, which implies that

$$\begin{aligned}
&\log \mathbb{E}(e^{\frac{1}{2}\lambda_1 Y_k}) + \log \mathbb{E}(e^{-\frac{1}{2}\lambda_1 X_k}) \\
&= \log\left(\sqrt{\alpha_1\beta_1} + \sqrt{(1-\alpha_1)(1-\beta_1)}\right)^2 \\
&= 2\log\left(\sqrt{\alpha_1\beta_1} + \left[1 - \frac{\alpha_1}{2} + \mathcal{O}(\alpha_1^2)\right]\left[1 - \frac{\beta_1}{2} + \mathcal{O}(\beta_1^2)\right]\right) \quad (34) \\
&= 2\log\left(1 - \left[\frac{1}{2}\alpha_1 + \frac{1}{2}\beta_1 - \sqrt{\alpha_1\beta_1} + \mathcal{O}(\alpha_1^2)\right]\right) \\
&= -I_1 \frac{\log n}{n} + \mathcal{O}(\alpha_1^2), \quad (35)
\end{aligned}$$

where $I_1 = \frac{n}{\log n}(\sqrt{\alpha_1} - \sqrt{\beta_1})^2$ by definition. (34) holds since $\sqrt{1-x} = 1 - \frac{1}{2}x + \mathcal{O}(x^2)$ for $x \to 0$, and (35) follows from Taylor series expansion. On the other hand,

$$\begin{aligned}
\log \mathbb{E}(e^{-\frac{1}{2}A_{ij}^{ab}}) &= \log\left(1 - p + p\sum_{z \in \mathcal{Z}} Q_{ab}(z) e^{-\frac{1}{2}\log\frac{Q_{ab}(z)}{Q_{\bar{a}b}(z)}}\right) \\
&= \log\left(1 - p \cdot H^2(Q_{ab}, Q_{\bar{a}b})\right) \\
&= -p \cdot H^2(Q_{ab}, Q_{\bar{a}b}) + \mathcal{O}(p^2), \quad (36)
\end{aligned}$$

where $H^2(\cdot,\cdot)$ is the square Hellinger distance, and $p = \Theta((\log n)/m) = o(1)$ since it is assumed that $m = \omega(\log n)$.

$$\mathbb{P}\left[\lambda_1\left(\sum_{k \in \mathcal{U}_{\bar{a}}^{(0)} \setminus \mathcal{U}_a} Y_k + \sum_{k \in \mathcal{S}_{a\bar{a}}} X_k - \sum_{k \in \mathcal{S}_{aa}} X_k - \sum_{k \in \mathcal{U}_a^{(0)} \setminus \mathcal{U}_a} Y_k\right) - \sum_{b \in [k_2]}\left[\sum_{j \in \mathcal{T}_{bb}} A_{ij}^{ab} + \sum_{\bar{b} \neq b}\sum_{j \in \mathcal{T}_{\bar{b}b}} \bar{A}_{ij}^{ab,\bar{b}}\right] \geq -(\epsilon/2)\log n\right] \quad (25)$$

$$\leq e^{\frac{\epsilon}{4}\log n} \cdot \mathbb{E}\left[\exp\left\{\sum_{k \in \mathcal{U}_{\bar{a}}^{(0)} \setminus \mathcal{U}_a} \frac{\lambda_1}{2} Y_k + \sum_{k \in \mathcal{S}_{a\bar{a}}} \frac{\lambda_1}{2} X_k - \sum_{k \in \mathcal{S}_{aa}} \frac{\lambda_1}{2} X_k - \sum_{k \in \mathcal{U}_a^{(0)} \setminus \mathcal{U}_a} \frac{\lambda_1}{2} Y_k - \sum_{b \in [k_2]}\left[\sum_{j \in \mathcal{T}_{bb}} \frac{A_{ij}^{ab}}{2} + \sum_{\bar{b} \neq b}\sum_{j \in \mathcal{T}_{\bar{b}b}} \frac{\bar{A}_{ij}^{ab,\bar{b}}}{2}\right]\right\}\right] \quad (26)$$

$$\begin{aligned}
&= e^{\frac{\epsilon}{4}\log n} \cdot \left[\prod_{k \in \mathcal{U}_{\bar{a}}^{(0)} \setminus \mathcal{U}_a} \mathbb{E}(e^{\frac{1}{2}\lambda_1 Y_k})\right] \cdot \left[\prod_{k \in \mathcal{S}_{a\bar{a}}} \mathbb{E}(e^{\frac{1}{2}\lambda_1 X_k})\right] \cdot \left[\prod_{k \in \mathcal{S}_{aa}} \mathbb{E}(e^{-\frac{1}{2}\lambda_1 X_k})\right] \cdot \left[\prod_{k \in \mathcal{U}_a^{(0)} \setminus \mathcal{U}_a} \mathbb{E}(e^{-\frac{1}{2}\lambda_1 Y_k})\right] \\
&\qquad \times \prod_{b \in [k_2]}\left[\prod_{j \in \mathcal{T}_{bb}} \mathbb{E}\left(e^{-\frac{1}{2}A_{ij}^{ab}}\right)\right] \cdot \left[\prod_{\bar{b} \neq b}\prod_{j \in \mathcal{T}_{\bar{b}b}} \mathbb{E}\left(e^{-\frac{1}{2}\bar{A}_{ij}^{ab,\bar{b}}}\right)\right] \quad (27)
\end{aligned}$$

$$\begin{aligned}
&= \exp\left\{\frac{\epsilon}{4}\log(n) + |\mathcal{U}_{\bar{a}}^{(0)} \setminus \mathcal{U}_a| \cdot \log \mathbb{E}(e^{\frac{1}{2}\lambda_1 Y_k}) + |\mathcal{S}_{a\bar{a}}| \cdot \log \mathbb{E}(e^{\frac{1}{2}\lambda_1 X_k}) + |\mathcal{S}_{aa}| \cdot \log \mathbb{E}(e^{-\frac{1}{2}\lambda_1 X_k})\right. \\
&\qquad \left. + |\mathcal{U}_a^{(0)} \setminus \mathcal{U}_a| \cdot \log \mathbb{E}(e^{-\frac{1}{2}\lambda_1 Y_k}) + \sum_{b \in [k_2]} |\mathcal{T}_{bb}| \cdot \times \log \mathbb{E}\left(e^{-\frac{1}{2}A_{ij}^{ab}}\right) + \sum_{\bar{b} \neq b} |\mathcal{T}_{\bar{b}b}| \log \mathbb{E}\left(e^{-\frac{1}{2}\bar{A}_{ij}^{ab,\bar{b}}}\right)\right\}. \quad (28)
\end{aligned}$$

Therefore, we bound the exponent in (28) from above in (32)–(33) shown at the bottom of this page, where (33) follows from (29)–(31), (35), and (36).

Next, one can show that

$$\left|\log\mathbb{E}(e^{\frac{1}{2}\lambda_1 X_k})\right|=\left|\log\left[1-\alpha_1+\alpha_1\sqrt{\frac{(1-\beta_1)\alpha_1}{(1-\alpha_1)\beta_1}}\right]\right|=\mathcal{O}\left(\frac{\log n}{n}\right),$$

$$\left|\log\mathbb{E}(e^{-\frac{1}{2}\lambda_1 Y_k})\right|=\left|\log\left[1-\beta_1+\beta_1\sqrt{\frac{(1-\alpha_1)\beta_1}{(1-\beta_1)\alpha_1}}\right]\right|=\mathcal{O}\left(\frac{\log n}{n}\right),$$

$$\left|\log\mathbb{E}(e^{-\frac{1}{2}\bar{A}_{ij}^{ab,\bar{b}}})\right|=\left|\log\left[1-p+p\cdot\sum_{z\in\mathcal{Z}}Q_{a\bar{b}}(z)e^{-\frac{1}{2}\log\frac{Q_{ab}(z)}{Q_{\bar{a}b}(z)}}\right]\right|$$

$$=\left|\log\left(1-p+p\cdot\sum_{z\in\mathcal{Z}}Q_{a\bar{b}}(z)\sqrt{\frac{Q_{\bar{a}b}(z)}{Q_{ab}(z)}}\right)\right|=\mathcal{O}(p).$$

Thus, the second line of (33) scales as $\mathcal{O}(\eta_n\log n)$, while the third line of (33) scales as $\mathcal{O}(\gamma_n mp)$. Therefore, we further bound (33) from above in (37)–(39) shown at the bottom of this page, where (38) holds since the sample probability $p=\frac{[(1+\epsilon)-\frac{I_1}{k_1}]\log n}{md_{\mathcal{U}}/k_2}$ and $d_{\mathcal{U}}\leq\sum_{b\in[k_2]}H^2(Q_{ab},Q_{\bar{a}b})$. Note that the last inequality holds for sufficiently large n, since all of the last four terms grows slower than $\Theta(\log n)$, and their sum is less than $\frac{\epsilon}{4}\log n$ for large enough n.

Therefore, we have $\mathbb{P}[\widetilde{L}_a(i)-\widetilde{L}_{\bar{a}}(i)\leq(\epsilon/2)\log n]\leq n^{-(1+\frac{\epsilon}{2})}$. Finally, by taking a union bound over all the clusters $\mathcal{U}_{\bar{a}}$ such that $\bar{a}\in[k_1]\setminus\{a\}$, we obtained that with probability at least $1-k_1n^{-(1+\frac{\epsilon}{2})}$, $\widetilde{L}_a(i)>\max_{\bar{a}\in[k_1]\setminus\{a\}}\widetilde{L}_{\bar{a}}(i)+(\epsilon/2)\log n$. This completes the proof of Lemma 4. □

Note that Lemma 4 is for a specific user $i\in[n]$. Taking a union bound over the n users yields that with probability $1-o(1)$, all the users $i\in[n]$ satisfy

$$\widetilde{L}_{\sigma(i)}(i)>\max_{\bar{a}\in[k_1]\setminus\{\sigma(i)\}}\widetilde{L}_{\bar{a}}(i)+(\epsilon/2)\log n, \tag{40}$$

where $\sigma(i)$ is the user cluster that user i belongs to.

Finally, it is shown in Lemma 5 below that the difference between the exact likelihood function $\widetilde{L}_a(i)$ and the original likelihood function $L_a(i)$ is negligible.

Lemma 5: With probability $1-o(1)$, there exists a sequence $\xi_n\in\Omega(\max\{\varepsilon_n,\varepsilon_n'\})\cap o(1)$ such that for all $a\in[k_1]$ and all users $i\in[n]$, $\left|L_a(i)-\widetilde{L}_a(i)\right|\leq\xi_n\log n$.

The proof of Lemma 5 can be found in Appendix VII-C. Combining (40) and Lemma 5 via the triangle inequality, we have that all the users satisfy $L_{\sigma(i)}(i)>\max_{\bar{a}\in[k_1]\setminus\{\sigma(i)\}}L_{\bar{a}}(i)$. This ensures the success of Stage 3, i.e., $\widehat{\sigma}(i)=\sigma(i),\forall i\in[n]$.

$$\begin{aligned}
&\frac{\epsilon}{4}\log n+\left[\frac{n}{k_1}-\eta_n n\right]\cdot\left(\log\mathbb{E}(e^{\frac{1}{2}\lambda_1 Y_k})+\log\mathbb{E}(e^{-\frac{1}{2}\lambda_1 X_k})\right)+\left[|\mathcal{U}_{\bar{a}}^{(0)}\setminus\mathcal{U}_a|-\frac{n}{k_1}+\eta_n n\right]\log\mathbb{E}(e^{\frac{1}{2}\lambda_1 Y_k})\\
&\quad+\left[|\mathcal{S}_{aa}|-\frac{n}{k_1}+\eta_n n\right]\log\mathbb{E}(e^{-\frac{1}{2}\lambda_1 X_k})+|\mathcal{S}_{a\bar{a}}|\log\mathbb{E}(e^{\frac{1}{2}\lambda_1 X_k})+|\mathcal{U}_a^{(0)}\setminus\mathcal{U}_a|\log\mathbb{E}(e^{-\frac{1}{2}\lambda_1 Y_k})\\
&\quad+\sum_{b\in[k_2]}\left[\frac{m}{k_2}-\gamma_n m\right]\log\mathbb{E}\left(e^{-\frac{1}{2}A_{ij}^{ab}}\right)+\left[|\mathcal{T}_{bb}|-\frac{m}{k_2}+\gamma_n m\right]\log\mathbb{E}\left(e^{-\frac{1}{2}A_{ij}^{ab}}\right)+\sum_{\bar{b}\neq b}|\mathcal{T}_{\bar{b}b}|\log\mathbb{E}\left(e^{-\frac{1}{2}\bar{A}_{ij}^{ab,\bar{b}}}\right) \qquad (32)\\
&\leq\frac{\epsilon}{4}\log n+\left[\frac{n}{k_1}-\eta_n n\right]\left(-I_1\frac{\log n}{n}+\mathcal{O}(\alpha_1^2)\right)+\sum_{b\in[k_2]}\left[\frac{m}{k_2}-\gamma_n m\right]\left(-p\cdot H^2(Q_{ab},Q_{\bar{a}b})+\mathcal{O}(p^2)\right)\\
&\quad+\eta_n n\left(\left|\log\mathbb{E}(e^{\frac{1}{2}\lambda_1 Y_k})\right|+\left|\log\mathbb{E}(e^{-\frac{1}{2}\lambda_1 X_k})\right|+\left|\log\mathbb{E}(e^{\frac{1}{2}\lambda_1 X_k})\right|+\left|\log\mathbb{E}(e^{-\frac{1}{2}\lambda_1 Y_k})\right|\right)\\
&\quad+\gamma_n m\left(\sum_{b\in[k_2]}\left|\log\mathbb{E}\left(e^{-\frac{1}{2}A_{ij}^{ab}}\right)\right|+\sum_{\bar{b}\neq b}\left|\log\mathbb{E}\left(e^{-\frac{1}{2}\bar{A}_{ij}^{ab,\bar{b}}}\right)\right|\right). \qquad (33)
\end{aligned}$$

$$\frac{\epsilon}{4}\log n-\frac{I_1}{k_1}\log n+\mathcal{O}(\eta_n\log n)+\mathcal{O}\left(\frac{(\log n)^2}{n}\right)-\left(\frac{mp}{k_2}\sum_{b\in[k_2]}H^2(Q_{ab},Q_{\bar{a}b})\right)+\mathcal{O}(\gamma_n mp)+\mathcal{O}(mp^2) \tag{37}$$

$$\leq\frac{\epsilon}{4}\log n-\frac{I_1}{k_1}\log n+\mathcal{O}(\eta_n\log n)+\mathcal{O}\left(\frac{(\log n)^2}{n}\right)-\left[(1+\epsilon)-\frac{I_1}{k_1}\right]\log n+\mathcal{O}(\gamma_n mp)+\mathcal{O}(mp^2) \tag{38}$$

$$=-\left(1+\frac{3}{4}\epsilon\right)\log n+\mathcal{O}(\eta_n\log n)+\mathcal{O}\left(\frac{(\log n)^2}{n}\right)+\mathcal{O}(\gamma_n mp)+\mathcal{O}(mp^2)\leq-\left(1+\frac{1}{2}\epsilon\right)\log n. \tag{39}$$

D. Analysis of Stage 4

The analysis of Stage 4 is similar to that of Stage 3. Lemma 6 below states that all the m items can be classified into the correct cluster when mnp satisfies (41).

Lemma 6: Suppose $\mathbf{h} \in \mathcal{G}$ and $(\sigma^{(0)}, \tau^{(0)}) \in \mathcal{G}'$. If

$$mnp \geq \frac{((1+\epsilon) - (I_2/k_2))m \log m}{d_{\mathcal{I}}/k_1}, \tag{41}$$

with probability $1 - o(1)$, all the items $j \in [m]$ satisfy $L'_{\tau(j)}(j) > \max_{\bar{b} \in [k_2] \setminus \{\tau(j)\}} L'_{\bar{b}}(j)$.

Finally, based on the outputs $\{\widehat{\mathcal{U}}_a\}_{a\in[k_1]}$ and $\{\widehat{\mathcal{I}}_b\}_{b\in[k_2]}$ of MC2G, one can recover the nominal matrix $\widehat{\mathsf{N}}$ via *majority voting*. Specifically, for $a \in [k_1]$ and $b \in [k_2]$, we define $u_{ab} := \arg\max_{z\in\mathcal{Z}} \sum_{i\in\widehat{\mathcal{U}}_a} \sum_{j\in\widehat{\mathcal{I}}_b} \mathbb{1}\{\mathsf{U}_{ij} = z\}$, and we then set

$$\widehat{\mathsf{N}}_{ij} = u_{ab}, \quad \text{if } i \in \widehat{\mathcal{U}}_a, j \in \widehat{\mathcal{I}}_b. \tag{42}$$

The correctness of (42) follows from the fact that $\sum_{i\in\widehat{\mathcal{U}}_a} \sum_{j\in\widehat{\mathcal{I}}_b} \mathbb{1}\{\mathsf{U}_{ij} = z\} \approx mnQ_{ab}(z)/(k_1k_2)$.

E. The Overall Success Probability

Let E_{suc} be the event that MC2G exactly recovers the nominal matrix. From the analyses of Stages 2–4, we know that for any $\mathbf{h} \in \mathcal{G}$ and $(\sigma^{(0)}, \tau^{(0)}) \in \mathcal{G}'$,

$$\mathbb{P}(E_{\text{suc}}|\mathbf{h}, (\sigma^{(0)}, \tau^{(0)})) \geq 1 - o(1), \tag{43}$$

where (43) is uniform in $\mathbf{h} \in \mathcal{G}$ and $(\sigma^{(0)}, \tau^{(0)}) \in \mathcal{G}'$. Therefore, the overall success probability satisfies

$$\begin{aligned}
\mathbb{P}(E_{\text{suc}}) &= \sum_{\mathbf{h}\in\mathcal{G}} \mathbb{P}(\mathbf{h})\mathbb{P}(E_{\text{suc}}|\mathbf{h}) + \sum_{\mathbf{h}\in\mathcal{B}_{\mathbf{h}}} \mathbb{P}(\mathbf{h})\mathbb{P}(E_{\text{suc}}|\mathbf{h}) \\
&\geq \sum_{\mathbf{h}\in\mathcal{G}} \mathbb{P}(\mathbf{h}) \sum_{(\sigma^{(0)},\tau^{(0)})\in\mathcal{G}'} \mathbb{P}((\sigma^{(0)}, \tau^{(0)})|\mathbf{h}) \\
&\qquad \times \mathbb{P}(E_{\text{suc}}|\mathbf{h}, (\sigma^{(0)}, \tau^{(0)})) \\
&\geq (1 - o(1)) \sum_{\mathbf{h}\in\mathcal{G}} \mathbb{P}(\mathbf{h}) \sum_{(\sigma^{(0)},\tau^{(0)})\in\mathcal{G}'} \mathbb{P}((\sigma^{(0)}, \tau^{(0)})|\mathbf{h}) \qquad (44) \\
&\geq (1 - o(1))(1 - \sqrt{\epsilon_n})^2(1 - 2\sqrt{\epsilon_n})^2 \\
&= (1 - o(1)), \qquad (45)
\end{aligned}$$

where (44) is due to (43), and (45) follows from (20) and Lemma 1.

VI. Proof Sketch of Theorem 2

The proof techniques used for Theorem 2 is a generalization of the techniques used in [12, Section IV-B], thus we only provide a proof sketch here. The key idea is to first show that the ML estimator ϕ_{ML} is the optimal estimator (as proved in [12, (33)]), and then analyze the error probability with respect to ϕ_{ML}—the crux of the analysis is to focus on a subset of of events that are most likely to induce errors, and to prove the tightness of the Chernoff bound.

To analyze ϕ_{ML}, we first show that under the model parameter $(\sigma, \tau, \mathsf{N})$ (where a single parameter ξ is used to be the abbreviation of $(\sigma, \tau, \mathsf{N})$ in the following), the log-likelihood of observing (U, G_1, G_2) is

$$\begin{aligned}
\log \mathbb{P}_\xi(\mathsf{U}, G_1, G_2) = e_1^\sigma \log \frac{\beta_1(1-\alpha_1)}{\alpha_1(1-\beta_1)} + e_2^\tau \log \frac{\beta_2(1-\alpha_2)}{\alpha_2(1-\beta_2)} \\
+ \sum_{a\in[k_1]} \sum_{b\in[k_2]} \sum_{z\in\mathcal{Z}} |\mathcal{D}_{ab}^z(\xi)| \cdot \log Q_{ab}(z) + C_0, \qquad (46)
\end{aligned}$$

where e_1^σ is the number of inter-cluster edges in G_1 with respect to σ; e_2^τ is the number of inter-cluster edges in G_2 with respect to τ; $\mathcal{D}_{ab}^z(\xi) = \{(i,j) \in [n] \times [m] : \sigma(i) = a, \tau(j) = b, \mathsf{U}_{ij} = z\}$ is the number of observed ratings z corresponding to user cluster $\mathcal{U}_a$ and item cluster $\mathcal{I}_b$; and C_0 is a constant that is independent of $(\sigma, \tau, \mathsf{N})$.

Suppose ξ is the ground truth that governs the model from now on, and note that the ML estimator ϕ_{ML} succeeds if ξ is the most likely model parameter in Ξ conditioned on the observation (U, G_1, G_2), i.e., $\log \mathbb{P}_\xi(\mathsf{U}, G_1, G_2)$ is larger than any other $\log \mathbb{P}_{\xi'}(\mathsf{U}, G_1, G_2)$ for $\xi' \in \Xi \setminus \{\xi\}$. In fact, what we show in the converse proof is that when mnp is less than the bound in (17), with high probability there exists another model parameter $\xi' \in \Xi \setminus \{\xi\}$ such that the likelihood $\log \mathbb{P}_{\xi'}(\mathsf{U}, G_1, G_2)$ achieves the maximum.

Specifically, let $\xi' \neq \xi$ be a model parameter that is identical to ξ except that its first component σ' differs from σ by only two labels, i.e., $\sum_{i\in[n]} \mathbb{1}\{\sigma'(i) \neq \sigma(i)\} = 2$. As the distinction between ξ' and ξ is small, the probability that $\log \mathbb{P}_{\xi'}(\mathsf{U}, G_1, G_2) \geq \log \mathbb{P}_\xi(\mathsf{U}, G_1, G_2)$ turns out to be relatively large, which is at least

$$\frac{1}{4} \exp\left\{-(1+o(1))\frac{2I_1(\log n)}{k_1} - (1+o(1))\frac{2mpd_{\mathcal{U}}}{k_2}\right\} \tag{47}$$

due to the tightness of the Chernoff bound (which can be proved by generalizing [12, Lemma 2]). In fact, one can find a subset $\Xi_0 \subseteq \Xi$ of model parameters such that $|\Xi_0| = \Theta(n)$ and each element in $\xi_0 \in \Xi_0$ satisfies (47) (i.e., the probability that ξ_0 induces an error is relatively large). This, together with the assumption that $mnp < k_2[\frac{1-\epsilon}{2} - \frac{I_1}{k_1}]n \log n/d_{\mathcal{U}}$, eventually implies that with probability approaching one, there exists at least one $\xi_0 \in \Xi_0$ such that $\log \mathbb{P}_{\xi_0}(\mathsf{U}, G_1, G_2) \geq \log \mathbb{P}_\xi(\mathsf{U}, G_1, G_2)$. Thus, the ML estimator fails.

In a similar and symmetric fashion, one can show that the ML estimator fails with probability approaching one, when $mnp < k_1[\frac{1-\epsilon}{2} - \frac{I_2}{k_2}]m \log m/d_{\mathcal{I}}$. This completes the proof of the converse part.

VII. Experiments

In this section, we apply the simplified version of MC2G mentioned in Remark 3 (without the information splitting step), as the sizes of the graphs m and n cannot be made arbitrarily large in the experiments.[5] That is, the four stages are applied to

[5] As discussed in Remark 3, the information splitting method is merely for the purpose of analysis, and the first part of the graphs $(G_1^{\text{a}}, G_2^{\text{a}})$ turns out to be too sparse to achieve weak recovery of clusters when m and n are not large enough.

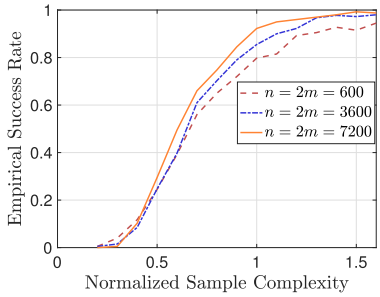


Fig. 3. The empirical success rate (over 400 trials) vs. the normalized sample complexity under the setting described in Section VII-A.

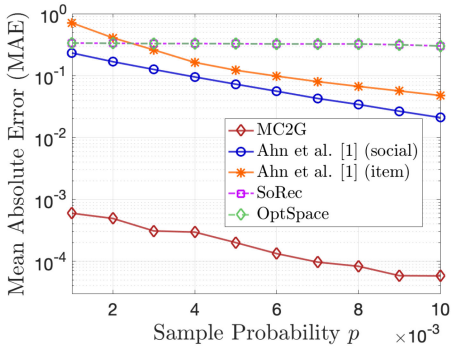


Fig. 4. Comparisons of MAEs of different algorithms under the synthetic setting described in Section VII-B.

the *fully-observed* graphs (G_1, G_2). While this implementation is slightly different from the original algorithm as described in Algorithm 1, its empirical performance nonetheless demonstrates a keen agreement with the theoretical guarantee for the original MC2G in Theorem 1 (as shown in Section VII-A below).

A. Verification of Theorem 1 on Synthetic Data

We verify the theoretical guarantee provided in Theorem 1 on a synthetic dataset generated according to a symmetric setting described as follows. The setting contains $k_1 = 3$ user clusters, $k_2 = 4$ item clusters, nominal ratings

$$z_{11} = 5, \quad z_{12} = 1, \quad z_{13} = 4, \quad z_{14} = 2,$$
$$z_{21} = 2, \quad z_{22} = 4, \quad z_{23} = 5, \quad z_{24} = 1,$$
$$z_{31} = 3, \quad z_{32} = 2, \quad z_{33} = 5, \quad z_{34} = 5,$$

with $\mathcal{Z} = \{1, 2, 3, 4, 5\}$, and the personalization distribution $Q_{V|Z}(v|z)$ that equals either 0.6 (if $v = z$) or 0.1 (if $v \neq z$). We set $n = 2\,m$, and both I_1 and I_2 (the qualities of social and item graphs) to 2. Fig. 3 shows the *empirical success rate* as a function of the *normalized sample complexity* for three different values of m and n. The empirical success rate is averaged over 400 random trials, and the normalized sample complexity is defined (according to Theorem 1) as mnp divided by

$$\max\left\{\frac{(1-(I_1/k_1))n\log n}{d_{\mathcal{U}}/k_2}, \frac{(1-(I_2/k_2))m\log m}{d_{\mathcal{I}}/k_1}\right\}. \quad (48)$$

It can be seen from Fig. 3 that as the normalized sample complexity increases, the empirical success rate also increases and becomes close to one when the normalized sample complexity exceeds one (corresponding to the success condition).

B. Comparing Mc2g With Other Algorithms on Synthetic Data

Next, we compare MC2G to several existing matrix completion algorithms on another synthetic dataset. The competitors include OPTSPACE [15] (a state-of-the-art matrix completion algorithm based on singular value decomposition followed by local manifold optimization), SoRec [3] (a matrix factorization based algorithm that incorporates social graphs), and a spectral clustering method with local refinements using *only* the social graph or *only* the item graph as side information by Ahn *et al.* [1]. This synthetic dataset is simpler compared to the one in Section VII-A, as we need to choose the ratings $\mathcal{Z}$ to be *binary* (as other competing algorithms are amenable only to binary ratings). It contains $n = 3000$ users partitioned into two user clusters, $m = 3000$ items partitioned into three item clusters, and we set the qualities of graphs $I_1 = 1.5$ and $I_2 = 2$, as well as the nominal ratings to be $z_{11} = 0$, $z_{12} = 1$, $z_{13} = 0$, $z_{21} = 0$, $z_{22} = 0$, $z_{23} = 1$. The personalization distributions are modelled as additive Bern(0.25) noise, i.e., $Q_{V|Z}(v|z)$ equals 0.75 if $v = z$, and equals 0.25 otherwise.

To ensure that the comparisons are fair, we quantize the outputs of the other algorithms to be $\{0, 1\}$-valued. We measure the performances using the *mean absolute error* (MAE)

$$\text{MAE} := \sum_{i=1}^{n}\sum_{j=1}^{m}\frac{|\widehat{\mathsf{N}}_{ij} - \mathsf{N}_{ij}|}{mn}. \quad (49)$$

Fig. 4 shows the MAE (averaged over 100 random trials) of each algorithm when $p \in [0.001, 0.01]$. It is clear that MC2G is orders of magnitude better than the competing algorithms in terms of the MAEs for this synthetic dataset.

C. Comparing Mc2g With Other Algorithms on Real Graphs

Next, we applied MC2G to a semi-real dataset that contains synthetic ratings but is inspired by real graphs.

- We adopt the LastFM social network [16] (collected in March 2020) as the social graph. Each node is a LastFM user, while each edge represents mutual follower relationships between users. The network contains $n = 7624$ users that are partitioned into 18 clusters with maximal size 1572 and minimal size 16. The sizes of all the 18 clusters and the empirical connection probabilities are provided in the supplementary material.
- We adopt the political blogs network [17] as the item similarity graph. Each node represents a blog that is either liberal-leaning or conservative-leaning, and each edge represents a link between two blogs. This network contains $m = 1222$ blogs which are partitioned into two clusters

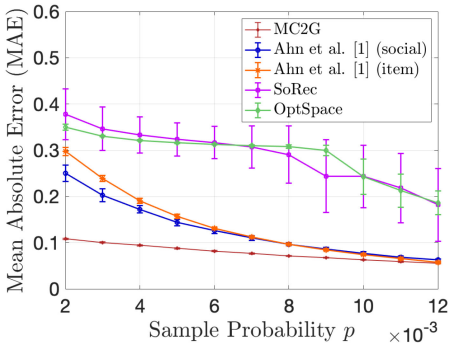


Fig. 5. Comparisons of MAEs of different algorithms under the semi-real setting described in Section VII-C, where we adopt the LastFM social network and political blog networks as social and item similarity graphs respectively. The length of each errorbar above and below each data point represents the standard deviations across the 100 independent trials.

with sizes $(m_1, m_2) = (586, 636)$, and the empirical connection probabilities are $\mathsf{B}' = \begin{bmatrix} 42.6 & 4.2 \\ 4.2 & 38.8 \end{bmatrix} \times 10^{-3}$.

We choose $\mathcal{Z} = \{0, 1\}$. The nominal ratings $\{z_{ij}\}$ for $i \in [18]$ and $j \in [2]$ are provided in the supplementary material. The personalization distributions are modelled as additive Bern(0.1) noise. The nominal matrix N is synthesized based on the user and item clusters as well as the nominal ratings. The personalized ratings matrix V is then synthesized from N and personalization distributions. Note that the objective is to recover the low rank nominal matrix based on partial observations of the personalized rating matrix V.

We compare MC2G to the algorithms introduced in VII-B on this semi-real dataset. Fig. 5 shows the MAE (averaged over 50 trials) of each algorithm when $p \in [0.002, 0.012]$. Clearly, MC2G is superior to the other algorithms, and the advantage is more significant when the sample probability p is small. In addition, the errorbars above and below each data point (representing one standard deviation) for MC2G are fairly small, demonstrating the statistical robustness of MC2G. The average running time (in seconds) of each algorithm, when $p = 0.01$, is as follows[6], showing that the running time of MC2G is commensurate with its prediction abilities.

MC2G	[1] (social)	[1] (item)	SoRec [3]	OPTSPACE [15]
$37.43s$	$26.24s$	$0.89s$	$0.47s$	$32.83s$

The reason why the running times of MC2G are longer than the algorithm in [1] is that MC2G performs spectral clustering for *both* social and item graphs, while their algorithm only performs spectral clustering for one graph. SoRec runs faster but its performance is rather poor, as can be seen from Figs. 4 and 5.

[6]We point out that MC2G, Ahn *et al.*'s algorithm [1], and SoRec are implemented in Python, while OPTSPACE is run in Matlab (since only its Matlab code is publicly available).

APPENDIX A
THEORETICAL GUARANTEES FOR WEAK RECOVERY

Theorem 3 (Adapted from Theorem 6 in [48]): Suppose an SBM contains N nodes that belong to K disjoint clusters with relative size $(p_1, \ldots, p_K)$, where $\sum_{k=1}^{K} p_k = 1$ and each p_k does not depend on N. Let $\delta : [N] \to [K]$ be the label function, R be the $K \times K$ symmetric connectivity matrix, and $\mathsf{R}_{\max} = \max_{i,j \in [K]} \mathsf{R}_{ij}$ be the largest probability.

Assume the following conditions hold: (i) $N\mathsf{R}_{\max} = \omega(1)$; (ii) there exists a constant $C_1 > 0$ such that for all $i, j, k \in [K]$, $\max\{\frac{\mathsf{R}_{ij}}{\mathsf{R}_{ik}}, \frac{1-\mathsf{R}_{ij}}{1-\mathsf{R}_{ik}}\} \le C_1$; (iii) there exists a constant $C_2 > 0$ such that $\sum_{k=1}^{K} (\mathsf{R}_{ik} - \mathsf{R}_{jk})^2 / \mathsf{R}_{\max}^2 \ge C_2$. Then, applying the spectral clustering method in [4, Algorithm 2] yields that with probability $1 - o(1)$, the estimated label function $\widehat{\delta}$ satisfies $l(\delta, \widehat{\delta}) = \mathcal{O}(\frac{(\log(N\mathsf{R}_{\max}))^2}{N\mathsf{R}_{\max}})$, where $l(\delta, \widehat{\delta}) := \min_{\pi \in \mathcal{S}_K} \frac{1}{N} \sum_{i \in [N]} \mathbb{1}\{\widehat{\delta}(i) \ne \pi(\delta(i))\}$ is the misclassification proportion.

We now check that the artificial SBM $\widetilde{G}_1$ satisfies the conditions in Theorem 3. Note that for $\widetilde{G}_1$, the entries in the connectivity matrix equal either $\alpha_1/\sqrt{\log n}$ or $\beta_1/\sqrt{\log n}$, both of which scale as $\Theta(\sqrt{\log n}/n)$ since α_1 and β_1 scale as $\Theta((\log n)/n)$. Thus, $n\alpha_1/\sqrt{\log n} = \Theta(\sqrt{\log n})$, and condition (i) holds. Also note that condition (ii) holds since α_1 and β_1 have the same scaling. Moreover, one can check that condition (iii) is equivalent to $2(\frac{\alpha_1}{\sqrt{\log n}} - \frac{\beta_1}{\sqrt{\log n}})^2 / (\frac{\alpha_1}{\sqrt{\log n}})^2 \ge C_2$, which clearly holds since α_1 and β_1 have the same scaling. Therefore, applying the spectral clustering method in [4, Algorithm 2] to $\widetilde{G}_1$ yields that with probability $1 - o(1)$,

$$l_1(\sigma^{(0)}, \sigma) = \mathcal{O}\left(\frac{(\log(n\alpha_1/\sqrt{\log n}))^2}{n\alpha_1/\sqrt{\log n}}\right) = \mathcal{O}\left(\frac{(\log\log n)^2}{\sqrt{\log n}}\right),$$

which tends to zero as n tends to infinity. Similarly, one can also show that applying the spectral clustering method in [4, Algorithm 2] to $\widetilde{G}_2$ yields that with probability $1 - o(1)$,

$$l_2(\tau^{(0)}, \tau) = \mathcal{O}\left(\frac{(\log(m\alpha_2/\sqrt{\log m}))^2}{m\alpha_2/\sqrt{\log m}}\right) = \mathcal{O}\left(\frac{(\log\log m)^2}{\sqrt{\log m}}\right),$$

which tends to zero as m tends to infinity (or equivalently, as n tends to infinity, since $n = \omega(\log m)$). Therefore, there exist vanishing sequences ϵ_n, η_n, and γ_n such that with probability at least $1 - \epsilon_n$, the spectral clustering method running on $\widetilde{G}_1$ and $\widetilde{G}_2$ ensure that $l_1(\sigma^{(0)}, \sigma) \le \eta_n$ and $l_2(\tau^{(0)}, \tau) \le \gamma_n$.

Remark 9: There is a subtle difference between the SBM considered in [48] and the current work. It is assumed in [48] that each node is assigned to the k-th cluster with probability p_k, thus when setting $(p_1, \ldots, p_K) = (1/K, \ldots, 1/K)$, all the clusters have size *approximately* n/K. In contrast, the current work assumes that all the clusters have *exactly* the same size of n/K. However, the theoretical guarantee of the spectral clustering method in [4, Algorithm 2] is valid for both models. The only difference in the proofs is that for the model in [48], one needs to additionally prove that the size of the k-th cluster is tightly concentrated around $p_k n$ (for all $k \in \{1, \ldots, K\}$) with high probability, by using proper concentration inequalities such

as the Chernoff bound. For example, inequality (31) in [48] holds only when the cluster size is at least $p_k n(1-o(1))$.

APPENDIX B
PROOF OF LEMMA 1

Consider the process of first generating a sub-graph $H_1^{\rm a}$ and then generating a sub-SBM $G_1^{\rm a}$ on the sub-graph $H_1^{\rm a}$. The probability that an edge $E_{ii'}$ (connecting nodes i and i') appears in $G_1^{\rm a}$ equals[7] $1/\sqrt{\log n}$ multiplied by α_1 or β_1 (depending on whether i and i' are in the same community). Thus, a key observation is that the aforementioned process is equivalent to generating $\widetilde{G}_1$ directly. By this observation and recalling that a spectral clustering method running on $\widetilde{G}_1$ ensures $l_1(\sigma^{(0)},\sigma)\le\eta_n$ with probability at least $1-\epsilon_n$ [48, Theorem 6], we have

$$\sum_{h_1^{\rm a}}\mathbb{P}(H_1^{\rm a}=h_1^{\rm a})P_{\rm suc}(h_1^{\rm a})\ge 1-\epsilon_n, \tag{50}$$

where $P_{\rm suc}(h_1^{\rm a})$ is the probability that a spectral clustering method running on $G_1^{\rm a}$ (which depends on $h_1^{\rm a}$) ensures $l_1(\sigma^{(0)},\sigma)\le\eta_n$. Let $\mathcal{H}_1^{{\rm a},\mathcal{G}}$ and $\mathcal{H}_1^{{\rm a},\mathcal{B}}$ respectively be the sets of good and bad sub-graphs $h_1^{\rm a}$. Suppose the probability of generating a good sub-graph (i.e., $h_1^{\rm a}\in\mathcal{H}_1^{{\rm a},\mathcal{G}}$) is less than $1-\sqrt{\epsilon_n}$. Then, by the definition of the good sub-graphs $h_1^{\rm a}$,

$$\begin{aligned}
&\sum_{h_1^{\rm a}}\mathbb{P}(H_1^{\rm a}=h_1^{\rm a})P_{\rm suc}(h_1^{\rm a})\\
&<\sum_{h_1^{\rm a}\in\mathcal{H}_1^{{\rm a},\mathcal{G}}}\mathbb{P}(H_1^{\rm a}=h_1^{\rm a})+\sum_{h_1^{\rm a}\in\mathcal{H}_1^{{\rm a},\mathcal{B}}}\mathbb{P}(H_1^{\rm a}=h_1^{\rm a})(1-\sqrt{\epsilon_n})\\
&=\sum_{h_1^{\rm a}\in\mathcal{H}_1^{{\rm a},\mathcal{G}}}\mathbb{P}(H_1^{\rm a}=h_1^{\rm a})+(1-\sqrt{\epsilon_n})\left(1-\sum_{h_1^{\rm a}\in\mathcal{H}_1^{{\rm a},\mathcal{G}}}\mathbb{P}(H_1^{\rm a}=h_1^{\rm a})\right)\\
&<1-\epsilon_n,
\end{aligned}$$

which yields a contradiction to (50). Thus, we conclude that with probability at least $1-\sqrt{\epsilon_n}$ over the generation of $H_1^{\rm a}$, the randomly generated $H_1^{\rm a}$ is a good sub-graph.

Now, we consider a specific user node $i\in[n]$, and use the random variable $D_{ii'}\sim{\rm Bern}(1/\sqrt{\log n})$ to denote whether or not the edge between nodes i and i' (where $i'\neq i$) belongs to $H_1^{\rm a}$. Let $D=\sum_{i'\neq i}D_{ii'}$ be the degree of node i in $H_1^{\rm a}$, and its expected value $\mathbb{E}(D)=(n-1)/\sqrt{\log n}$.

Theorem 4 (Multiplicative Chernoff bound): Suppose $X_1,\ldots,X_n$ are independent random variables taking values in $\{0,1\}$. Let $X:=\sum_{i=1}^n X_i$ denote their sum and $\mathbb{E}(X)$ denote the sum's expected value. Then, for any $\delta>0$, $\mathbb{P}(X\ge(1+\delta)\mathbb{E}(X))\le\exp\{-\frac{\delta^2\mathbb{E}(X)}{2+\delta}\}$.

Proof: The proof of Theorem 4 follows from [56, Ex. 4.7] together with the inequality $\frac{2\delta}{2+\delta}\le\log(1+\delta)$ for $\delta\ge 0$. □

Applying Theorem 4 with $\delta=\frac{2n}{n-1}-1$, we have

$$\mathbb{P}\left[D\ge\frac{2n}{\sqrt{\log n}}\right]=\mathbb{P}\left(D\ge(1+\delta)\mathbb{E}(D)\right)\le e^{-\frac{\delta^2(n-1)}{(2+\delta)\sqrt{\log n}}}.$$

[7]Specifically, the probability that an edge $E_{ii'}$ appears in $G_1^{\rm a}$ is equal to the probability of $E_{ii'}$ belonging to $H_1^{\rm a}$ multiplied by the probability of generating $E_{ii'}$ in the sub-SBM $G_1^{\rm a}$.

Thus, with probability at least $1-\exp\{-\frac{\delta^2(n-1)}{(2+\delta)\sqrt{\log n}}\}$, the degree of node i in $H_1^{\rm a}$ is at most $2n/\sqrt{\log n}$. A union bound over *all* user nodes guarantees that, with probability at least $1-n\exp\{-\frac{\delta^2(n-1)}{(2+\delta)\sqrt{\log n}}\}=1-\exp(-\Theta(n/\sqrt{\log n}))$, the degrees of *all* the nodes in $H_1^{\rm a}$ is at most $2n/\sqrt{\log n}$, which implies the complement sub-graph $H_1^{\rm b}$ is good.

Without loss of generality, we can assume that $\sqrt{\epsilon_n}$ decays slower than $\exp(-\Theta(n/\sqrt{\log n}))$, i.e., $\sqrt{\epsilon_n}>\exp(-\Theta(n/\sqrt{\log n}))$ for sufficiently large n. This is because we have the flexibility to choose ϵ_n: even if it is allowed to choose an ϵ_n such that $\sqrt{\epsilon_n}$ decays faster than $\exp(-\Theta(n/\sqrt{\log n}))$, one can always decide to choose an ϵ_n that does not decay so fast. Thus, applying a union bound (over the "good events" for $H_1^{\rm a}$ and $H_1^{\rm b}$) implies that for sufficiently large n, the probability that *both* $H_1^{\rm a}$ and $H_1^{\rm b}$ are good is at least $1-2\sqrt{\epsilon_n}$. In a similar manner, we can also prove the analogous statements for $H_2^{\rm a}$ and $H_2^{\rm b}$. Due to the independence of the generations of $(H_1^{\rm a},H_1^{\rm b})$ and $(H_2^{\rm a},H_2^{\rm b})$, the probability that $H_1^{\rm a},H_1^{\rm b},H_2^{\rm a},H_2^{\rm b}$ are all good is at least $(1-2\sqrt{\epsilon_n})^2$.

APPENDIX C
PROOF OF LEMMA 2

We first analyze the estimates $\{\widehat{\mathsf{B}}_{aa}\}_{a\in[k_1]}$ in (5). By letting $\{X_k\}\overset{\text{i.i.d.}}{\sim}{\rm Bern}(\alpha_1)$ and $\{Y_k\}\overset{\text{i.i.d.}}{\sim}{\rm Bern}(\beta_1)$, we have $e(\mathcal{U}_a^{(0)},\mathcal{U}_a^{(0)})=\sum_{k=1}^{B_1}X_k+\sum_{k=1}^{B_2}Y_k$, where $B_1:=\sum_{\bar{a}\in[k_1]}\binom{|\mathcal{S}_{\bar{a}a}|}{2}$ and $B_2:=\binom{|\mathcal{U}_a^{(0)}|}{2}-B_1$. Note that $\mu_{\mathsf{B}_{aa}}:=\mathbb{E}[e(\mathcal{U}_a^{(0)},\mathcal{U}_a^{(0)})]\le\alpha_1\binom{|\mathcal{U}_a^{(0)}|}{2}$. On the other hand, since the degree of any nodes in $h_1^{\rm b}$ is at least $n(1-2/\sqrt{\log n})$ (or equivalently, the number of non-edges of any nodes is at most $2n/\sqrt{\log n}$), we know that

$$e(\mathcal{U}_a^{(0)},\mathcal{U}_a^{(0)})\ge\sum_{k=1}^{B_1-\frac{n}{2}\frac{2n}{\sqrt{\log n}}}X_k,\ \text{and}\ \mu_{\mathsf{B}_{aa}}\ge\left[B_1-\frac{n^2}{\sqrt{\log n}}\right]\alpha_1.$$

Applying Theorem 4 yields that for any $\delta\in(0,1)$, with probability at least $1-2\exp(-\delta^2\mu_{\mathsf{B}_{aa}}/3)$,

$$\begin{aligned}
(1-\delta)\left(B_1-\frac{n^2}{\sqrt{\log n}}\right)\alpha_1&\le e(\mathcal{U}_a^{(0)},\mathcal{U}_a^{(0)})\\
&\le(1+\delta)\alpha_1\binom{|\mathcal{U}_a^{(0)}|}{2}.
\end{aligned}$$

As the estimate $\widehat{\mathsf{B}}_{aa}=e(\mathcal{U}_a^{(0)},\mathcal{U}_a^{(0)})/\binom{|\mathcal{U}_a^{(0)}|}{2}$, we then have

$$\left(1-\delta-c_1\eta_n-\frac{c_2}{\sqrt{\log n}}\right)\alpha_1\le\widehat{\mathsf{B}}_{aa}\le(1+\delta)\,\alpha_1.$$

for some constants $c_1,c_2>0$. By choosing $\delta=1/\sqrt{\log n}$, we complete the proof for $\widehat{\mathsf{B}}_{aa}$.

The analyses of other estimates $\{\widehat{\mathsf{B}}_{aa'}\}_{a\neq a'}$, $\{\widehat{\mathsf{B}}_{bb}\}_{b\in[k_2]}$, and $\{\widehat{\mathsf{B}}_{bb'}\}_{b\neq b'}$ are similar, thus we omit them for brevity (except that we need to replace η_n by γ_n for $\{\widehat{\mathsf{B}}_{bb}\}_{b\in[k_2]}$, and $\{\widehat{\mathsf{B}}_{bb'}\}_{b\neq b'}$). Therefore, one can find a sequence $\varepsilon_n\in\Omega(\max\{\gamma_n,\eta_n,1/\sqrt{\log n}\})\cap o(1)$ such that Lemma 2 holds.

APPENDIX D
PROOF OF LEMMA 3

Let us recall the definition of $\widehat{Q}_{ab}(z)$ in (7), in which the numerator $|\mathcal{Q}^z_{ab}| = \sum_{i\in\mathcal{U}_a^{(0)}} \sum_{j\in\mathcal{I}_b^{(0)}} \mathbb{1}\{\mathsf{U}_{ij}=z\}$. Let $\{T_{ij}\} \overset{\text{i.i.d.}}{\sim} \text{Bern}(p)$, $\{Z^{ab}_{ij}\} \overset{\text{i.i.d.}}{\sim} Q_{ab}$ for all $a\in[k_1]$ and $b\in[k_2]$. Thus, $|\mathcal{Q}^z_{ab}|$ can be rewritten as

$$\sum_{i\in\mathcal{S}_{aa}} \sum_{j\in\mathcal{T}_{bb}} T_{ij}\mathbb{1}(Z^{ab}_{ij}=z) + \sum_{\bar{a}\neq a}\sum_{\bar{b}\neq b}\sum_{i\in\mathcal{S}_{\bar{a}a}}\sum_{j\in\mathcal{T}_{\bar{b}\bar{b}}} T_{ij}\mathbb{1}(Z^{\bar{a}\bar{b}}_{ij}=z).$$

Note that the number of summands in the first term

$$|i\in\mathcal{S}_{aa}|\cdot|j\in\mathcal{T}_{bb}| \geq [((n/k_1)-\eta_n n)\,((m/k_2)-\gamma_n m)] := L.$$

Thus, the expectation of $|\mathcal{Q}^z_{ab}|$ satisfies

$$\mathbb{E}(|\mathcal{Q}^z_{ab}|) \geq L\cdot\mathbb{E}(T_{ij}\mathbb{1}(Z^{ab}_{ij}=z)) \geq LpQ_{ab}(z), \quad \text{and}$$
$$\mathbb{E}(|\mathcal{Q}^z_{ab}|) \leq L\cdot\mathbb{E}(T_{ij}\mathbb{1}(Z_{ij}=z)) + (mn/(k_1k_2)-L)\mathbb{E}(T_{ij}),$$

where the upper bound is due to the fact that $\mathbb{1}(Z^{\bar{a}\bar{b}}_{ij}=z)\leq 1$. Applying Theorem 4 yields that with probability $1-\exp(-\Theta(\delta^2\mathbb{E}(|\mathcal{Q}^z_{ab}|)))$, for all $z\in\mathcal{Z}$,

$$|\mathcal{Q}^z_{ab}| \geq (1-\delta)LpQ_{ab}(z), \quad \text{and}$$
$$|\mathcal{Q}^z_{ab}| \leq (1+\delta)(1+\Theta(\max\{\eta_n,\gamma_n\}))LpQ_{ab}(z),$$

where $\delta\in(0,1)$. Choosing $\delta = 1/\sqrt{\log n}$, we ensure that with probability $1-o(1)$, for all $z\in\mathcal{Z}$, $a\in[k_1]$, and $b\in[k_2]$,

$$\left|(\widehat{Q}_{ab}(z)/Q_{ab}(z))-1\right| = \mathcal{O}(\max\{\eta_n,\gamma_n,1/\sqrt{\log n}\}).$$

APPENDIX E
PROOF OF LEMMA 5

First note that

$$\left|L_a(i)-\widetilde{L}_a(i)\right| \leq \sum_{a'\in[k_1]} e(\{i\},\mathcal{U}^{(0)}_{a'}) \left|\log\frac{\mathsf{B}_{aa'}(1-\widehat{\mathsf{B}}_{aa'})}{\widehat{\mathsf{B}}_{aa'}(1-\mathsf{B}_{aa'})}\right| + \sum_{b\in[k_2]}\sum_{j\in\mathcal{I}_b^{(0)}} \mathbb{1}\{\mathsf{U}_{ij}\neq\mathsf{e}\}\cdot\left|\log\frac{Q_{ab}(\mathsf{U}_{ij})}{\widehat{Q}_{ab}(\mathsf{U}_{ij})}\right|. \tag{51}$$

As $\sum_{a'\in[k_1]} e(\{i\},\mathcal{U}^{(0)}_{a'})$ represents the degree of user i in the social graph, and its expectation μ_i satisfies $n\beta_1\leq\mu_i\leq n\alpha_1$. By applying Theorem 4, we have that for any $\kappa>0$,

$$\mathbb{P}\left(\sum_{a'\in[k_1]} e(\{i\},\mathcal{U}^{(0)}_{a'}) \geq (1+\kappa)n\alpha_1\right) \leq e^{-\frac{\kappa^2}{2+\kappa}n\beta_1}. \tag{52}$$

We choose κ to be a large enough constant that ensures the RHS of (52) to scale as $o(n^{-1})$. Then, by applying the union bound over all the n users, we have that with probability $1-o(1)$, all the users $i\in[n]$ satisfy

$$\sum_{a'\in[k_1]} e(\{i\},\mathcal{U}^{(0)}_{a'}) \leq (1+\kappa)n\alpha_1 = c_3\log n, \tag{53}$$

for some constant $c_3>0$. Also, note that the term $\sum_{b\in[k_2]}\sum_{j\in\mathcal{I}_b^{(0)}}\mathbb{1}\{\mathsf{U}_{ij}\neq\mathsf{e}\}$ corresponds to the number of observed ratings for each user. By a similar analysis (based on the Chernoff bound), one can show that with probability $1-o(1)$, all the users $i\in[n]$ satisfy

$$\sum_{b\in[k_2]}\sum_{j\in\mathcal{I}_b^{(0)}} \mathbb{1}\{\mathsf{U}_{ij}\neq\mathsf{e}\} \leq c_4\log n, \tag{54}$$

for some constant $c_4>0$.

Recall from Lemmas 2 and 3 that the estimated connection probabilities satisfy $\left|(\widehat{\mathsf{B}}_{aa'}-\mathsf{B}_{aa'})/\mathsf{B}_{aa'}\right|\leq\varepsilon_n$ for all $a,a'\in[k_1]$, and the estimated personalization distribution $\left|(\widehat{Q}_{ab}(z)/Q_{ab}(z))-1\right|\leq\varepsilon'_n$ for all $a\in[k_1], b\in[k_2], z\in\mathcal{Z}$. By applying a Taylor series expansion, we then have

$$\left|\log\frac{\mathsf{B}_{aa'}(1-\widehat{\mathsf{B}}_{aa'})}{\widehat{\mathsf{B}}_{aa'}(1-\mathsf{B}_{aa'})}\right| \leq 2\varepsilon_n, \quad \left|\log\frac{Q_{ab}(\mathsf{U}_{ij})}{\widehat{Q}_{ab}(\mathsf{U}_{ij})}\right| \leq 2\varepsilon'_n. \tag{55}$$

Combining (53), (54), and (55), we complete the proof of Lemma 5.

REFERENCES

[1] K. Ahn, K. Lee, H. Cha, and C. Suh, "Binary rating estimation with graph side information," in *Proc. Neural Inf. Process. Syst.*, 2018, pp. 4272–4283.

[2] C. Jo and K. Lee, "Discrete-valued latent preference matrix estimation with graph side information," in *Proc. Int. Conf. Mach. Learn.*, 2021, pp. 5107–5117.

[3] H. Ma, H. Yang, M. R. Lyu, and I. King, "SoRec: Social recommendation using probabilistic matrix factorization," in *Proc. Int. Conf. Inf. Knowl. Manage.*, 2008, pp. 931–940.

[4] H. Ma, D. Zhou, C. Liu, M. R. Lyu, and I. King, "Recommender systems with social regularization," in *Proc. ACM Int. Conf. Web Search Data Mining*, 2011, pp. 287–296.

[5] D. Cai, X. He, J. Han, and T. S. Huang, "Graph regularized nonnegative matrix factorization for data representation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 8, pp. 1548–1560, Aug. 2011.

[6] V. Kalofolias, X. Bresson, M. Bronstein, and P. Vandergheynst, "Matrix completion on graphs," in *Proc. Neural Inf. Process. Syst. Workshop Out Box: Robustness High Dimens.*, 2014.

[7] T. Zhou, H. Shan, A. Banerjee, and G. Sapiro, "Kernelized probabilistic matrix factorization: Exploiting graphs and side information," in *Proc. SIAM Int. Conf. Data Min.*, 2012, pp. 403–414.

[8] N. Rao, H.-F. Yu, P. Ravikumar, and I. S. Dhillon, "Collaborative filtering with graph information: Consistency and scalable methods," in *Proc. Neural Inf. Process. Syst.*, 2015, pp. 2107–2115.

[9] D. Goldberg, D. Nichols, B. M. Oki, and D. Terry, "Using collaborative filtering to weave an information tapestry," *Commun. ACM*, vol. 35, no. 12, pp. 61–70, 1992.

[10] M. K. Condliff, D. D. Lewis, D. Madigan, and C. Posse, "Bayesian mixed-effects models for recommender systems," in *Proc. ACM SIGIR Workshop Recommender Syst.*, 1999, pp. 23–30.

[11] J. Wang, P. Huang, H. Zhao, Z. Zhang, B. Zhao, and D. L. Lee, "Billion-scale commodity embedding for E-commerce recommendation in Alibaba," in *Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Min.*, 2018, pp. 839–848.

[12] Q. Zhang, V. Y. F. Tan, and C. Suh, "Community detection and matrix completion with social and item similarity graphs," *IEEE Trans. Signal Process.*, vol. 69, pp. 917–931, 2021.

[13] J. Bennett and S. Lanning, "The Netflix prize," in *Proc. KDD Cup Workshop*, vol. 2007, 2007, p. 35.

[14] P. W. Holland, K. B. Laskey, and S. Leinhardt, "Stochastic blockmodels: First steps," *Social Netw.*, vol. 5, no. 2, pp. 109–137, 1983.

[15] R. H. Keshavan, A. Montanari, and S. Oh, "Matrix completion from a few entries," *IEEE Trans. Inf. Theory*, vol. 56, no. 6, pp. 2980–2998, Jun. 2010.

[16] B. Rozemberczki and R. Sarkar, "Characteristic functions on graphs: Birds of a feather, from statistical descriptors to parametric models," in *Proc. Int. Conf. Inf. Knowl. Manage.*, 2020, pp. 1325–1334.

[17] L. A. Adamic and N. Glance, "The political blogosphere and the 2004 US election: Divided they blog," in *Proc. 3rd Int. Workshop Link Discov.*, 2005, pp. 36–43.

[18] E. J. Candés and B. Recht, "Exact matrix completion via convex optimization," *Found. Comput. Math.*, vol. 9, no. 6, pp. 717–772, 2009.

[19] E. J. Candés and T. Tao, "The power of convex relaxation: Near-optimal matrix completion," *IEEE Trans. Inf. Theory*, vol. 56, no. 5, pp. 2053–2080, May 2010.

[20] G. Marjanovic and V. Solo, "On l_q optimization and matrix completion," *IEEE Trans. Signal Process.*, vol. 60, no. 11, pp. 5714–5724, Nov. 2012.

[21] S. Chen, A. Sandryhaila, J. M. Moura, and J. Kovačević, "Signal recovery on graphs: Variation minimization," *IEEE Trans. Signal Process.*, vol. 63, no. 17, pp. 4609–4624, Sep. 2015.

[22] W. Dai, O. Milenkovic, and E. Kerman, "Subspace evolution and transfer (SET) for low-rank matrix completion," *IEEE Trans. Signal Process.*, vol. 59, no. 7, pp. 3120–3132, Jul. 2011.

[23] R. Ma, N. Barzigar, A. Roozgard, and S. Cheng, "Decomposition approach for low-rank matrix completion and its applications," *IEEE Trans. Signal Process.*, vol. 62, no. 7, pp. 1671–1683, Apr. 2014.

[24] M. Jamali and M. Ester, "A matrix factorization technique with trust propagation for recommendation in social networks," in *Proc. 4th ACM Conf. Recommender Syst.*, 2010, pp. 135–142.

[25] G. Guo, J. Zhang, and N. Yorke-Smith, "TrustSVD: Collaborative filtering with both the explicit and implicit influence of user trust and of item ratings," in *Proc. 9th AAAI Conf. Artif. Intell.*, 2015, pp. 123–129.

[26] P. Massa and P. Avesani, "Controversial users demand local trust metrics: An experimental study on epinions.com community," in *Proc. AAAI Conf. Artif. Intell.*, 2005.

[27] X. Yang, H. Steck, Y. Guo, and Y. Liu, "On top-k recommendation using social networks," in *Proc. ACM Conf. Recommender Syst.*, 2012, pp. 67–74.

[28] F. Monti, M. Bronstein, and X. Bresson, "Geometric matrix completion with recurrent multi-graph neural networks," in *Proc. Neural Inf. Process. Syst.*, 2017, pp. 3700–3710.

[29] R. V. D. Berg, T. N. Kipf, and M. Welling, "Graph convolutional matrix completion," in *Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Min.*, Deep Learning Day, London, U.K., Aug. 2018, pp. 1–7.

[30] X. Wang, X. He, M. Wang, F. Feng, and T.-S. Chua, "Neural graph collaborative filtering," in *Proc. 42nd Int. ACM SIGIR Conf. Res. Dev. Inf. Retr.*, 2019, pp. 165–174.

[31] J. Yoon, K. Lee, and C. Suh, "On the joint recovery of community structure and community features," in *Proc. 56th Annu. Allert. Conf. Commun. Control Comput.*, 2018, pp. 686–694.

[32] E. Abbe, A. S. Bandeira, and G. Hall, "Exact recovery in the stochastic block model," *IEEE Trans. Inf. Theory*, vol. 62, no. 1, pp. 471–487, Jan. 2015.

[33] E. Mossel, J. Neeman, and A. Sly, "Reconstruction and estimation in the planted partition model," *Probab. Theory Related Fields*, vol. 162, pp. 431–461, 2015.

[34] E. Abbe and C. Sandon, "Community detection in general stochastic block models: Fundamental limits and efficient algorithms for recovery," in *Proc. IEEE Annu. Symp. Found. Comput. Sci.*, 2015, pp. 670–688.

[35] E. Abbe, "Community detection and stochastic block models: Recent developments," *J. Mach. Learn. Res.*, vol. 18, pp. 6446–6531, 2017.

[36] A. Y. Zhang and H. H. Zhou, "Minimax rates of community detection in stochastic block models," *Ann. Statist.*, vol. 44, pp. 2252–2280, 2016.

[37] B. Hajek, Y. Wu, and J. Xu, "Information limits for recovering a hidden community," *IEEE Trans. Inf. Theory*, vol. 63, no. 8, pp. 4729–4745, Aug. 2017.

[38] H. Saad and A. Nosratinia, "Community detection with side information: Exact recovery under the stochastic block model," *IEEE J. Sel. Top. Signal Process.*, vol. 12, no. 5, pp. 944–958, Oct. 2018.

[39] H. Saad and A. Nosratinia, "Exact recovery in community detection with continuous-valued side information," *IEEE Signal Process. Lett.*, vol. 26, no. 2, pp. 332–336, Feb. 2019.

[40] V. Mayya and G. Reeves, "Mutual information in community detection with covariate information and correlated networks," in *Proc. 57th Annu. Allert. Conf. Commun. Control Comput.*, 2019, pp. 602–607.

[41] A. R. Asadi, E. Abbe, and S. Verdú, "Compressing data on graphs with clusters," in *Proc. IEEE Int. Symp. Inf. Theory*, 2017, pp. 1583–1587.

[42] M. McPherson, L. Smith-Lovin, and J. M. Cook, "Birds of a feather: Homophily in social networks," *Annu. Rev. Sociol.*, vol. 27, pp. 415–444, 2001.

[43] P. Chin, A. Rao, and V. Vu, "Stochastic block model and community detection in sparse graphs: A spectral algorithm with optimal rate of recovery," in *Proc. Conf. Learn. Theory*, 2015, pp. 391–423.

[44] C. Gao, Z. Ma, A. Y. Zhang, and H. H. Zhou, "Achieving optimal misclassification proportion in stochastic block models," *J. Mach. Learn. Res.*, vol. 18, no. 1, pp. 1980–2024, 2017.

[45] A. Javanmard, A. Montanari, and F. Ricci-Tersenghi, "Phase transitions in semidefinite relaxations," *Proc. Nat. Acad. Sci.*, vol. 113, pp. E2218–E2223, 2016.

[46] E. Mossel and J. Xu, "Density evolution in the degree-correlated stochastic block model," in *Proc. Conf. Learn. Theory*, 2016, pp. 1319–1356.

[47] F. Krzakala *et al.*, "Spectral redemption in clustering sparse networks," *Proc. Nat. Acad. Sci.*, vol. 110, no. 52, pp. 20935–20940, 2013.

[48] S.-Y. Yun and A. Proutiere, "Optimal cluster recovery in the labeled stochastic block model," in *Proc. 30th Int. Conf. Neural Inf. Process. Syst.*, 2016, pp. 973–981.

[49] N. Halko, P.-G. Martinsson, and J. A. Tropp, "Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions," *SIAM Rev.*, vol. 53, no. 2, pp. 217–288, 2011.

[50] Y. Chen, A. Jalali, S. Sanghavi, and C. Caramanis, "Low-rank matrix recovery from errors and erasures," *IEEE Trans. Inf. Theory*, vol. 59, no. 7, pp. 4324–4337, Jul. 2013.

[51] E. J. Candes and Y. Plan, "Matrix completion with noise," *Proc. IEEE*, vol. 98, no. 6, pp. 925–936, Jun. 2010.

[52] S. Negahban and M. J. Wainwright, "Restricted strong convexity and weighted matrix completion: Optimal bounds with noise," *J. Mach. Learn. Res.*, vol. 13, no. 1, pp. 1665–1697, 2012.

[53] M. Zhang and Y. Chen, "Inductive matrix completion based on graph neural networks," in *Proc. Int. Conf. Learn. Representations*, 2020, pp. 1–25.

[54] B. Recht, "A simpler approach to matrix completion," *J. Mach. Learn. Res.*, vol. 12, pp. 3413–3430, Jan. 2011.

[55] Y. Chen, "Incoherence-optimal matrix completion," *IEEE Trans. Inf. Theory*, vol. 61, no. 5, pp. 2909–2923, May 2015.

[56] M. Mitzenmacher and E. Upfal, *Probability and Computing: Randomization and Probabilistic Techniques in Algorithms and Data Analysis*. Cambridge, U.K.: Cambridge Univ. Press, 2017.