Discrete-Valued Latent Preference Matrix Estimation with Graph Side Information

Changhun Jo[1] **Kangwook Lee**[2]

Abstract

Incorporating graph side information into recommender systems has been widely used to better predict ratings, but relatively few works have focused on theoretical guarantees. Ahn et al. (2018) firstly characterized the optimal sample complexity in the presence of graph side information, but the results are limited due to strict, unrealistic assumptions made on the unknown latent preference matrix and the structure of user clusters. In this work, we propose a new model in which 1) the unknown latent preference matrix can have any discrete values, and 2) users can be clustered into multiple clusters, thereby relaxing the assumptions made in prior work. Under this new model, we fully characterize the optimal sample complexity and develop a computationally-efficient algorithm that matches the optimal sample complexity. Our algorithm is robust to model errors and outperforms the existing algorithms in terms of prediction performance on both synthetic and real data.

1. Introduction

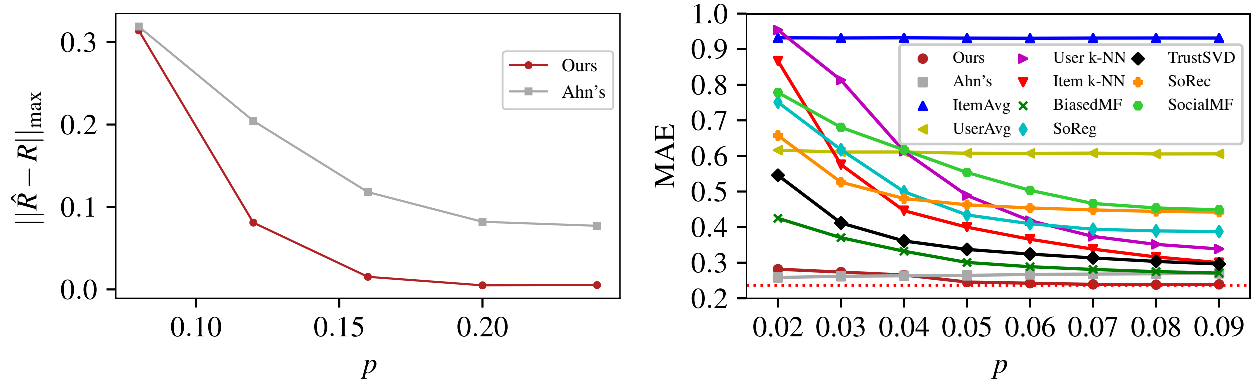
Recommender systems provide suggestions for items based on users' decisions such as ratings given to those items. Collaborative filtering is a popular approach to designing recommender systems (Herlocker et al., 1999; Sarwar et al., 2001; Linden et al., 2003; Rennie & Srebro, 2005; Salakhutdinov & Mnih, 2007; 2008; Agarwal & Chen, 2010; Davenport et al., 2014). However, collaborative filtering suffers from the well-known cold start problem since it relies only on past interactions between users and items. With the exponential growth of social media, recommender systems have started to use a social graph to resolve the cold start problem. For instance, Jamali & Ester (2010) provide an algorithm that handles the cold start problem by exploiting social graph information.

While a lot of works have improved the performance of algorithms by incorporating graph side information into recommender systems (Jamali & Ester, 2009a;b; 2010; Cai et al., 2011; Ma et al., 2011; Yang et al., 2012; 2013b; Kalofolias et al., 2014), relatively few works have focused on justifying theoretical guarantees of the performance (Chiang et al., 2015; Rao et al., 2015; Ahn et al., 2018). One notable exception is the recent work of Ahn et al. (2018), which finds the minimum number of observed ratings for reliable recovery of the latent preference matrix with social graph information and partial observation of the rating matrix. They also provide an efficient algorithm with low computational complexity. However, the assumptions made in this work are too strong to reflect the real-world data. In specific, they assume that each user rates each item either $+1$ (like) or -1 (dislike), and that the observations are noisy so that they can be flipped with probability $\theta \in (0, \frac{1}{2})$. This assumption can be interpreted as each user rates each item $+1$ with probability $1-\theta$ or θ. Note that this parametric model is very limited, so the discrepancy between the model and the real world occurs; if a user likes item a, b and c with probability $1/4$, $1/3$ and $3/4$ respectively, then the model cannot represent this case well (see Rmk. 3 for a detailed description).

This motivates us to propose a general model that better represents real data. Specifically, we assume that user i likes item j with probability R_{ij}, which we call user i's latent preference level on item j, and R_{ij} belongs to the discrete set $\{p_1, \ldots, p_d\}$ where $d \geq 1$ and $0 < p_1 < \cdots < p_d < 1$. As d can be any positive integer, our generalized model can reflect various preference levels on different items. In addition to that, we assume that the social graph information follows the Stochastic Block Model (SBM) (Holland et al., 1983), and the social graph is correlated with the latent preference matrix R in a specific way, which we will detail in Sec. 3. Under this highly generalized model, we fully characterize the optimal sample complexity required for estimating the latent preference matrix R. To the best of

[1] Department of Mathematics, University of Wisconsin-Madison, Madison, Wisconsin, USA [2] Department of Electrical and Computer Engineering, University of Wisconsin-Madison, Madison, Wisconsin, USA. Correspondence to: Changhun Jo <cjo4@wisc.edu>, Kangwook Lee <kangwook.lee@wisc.edu>.

Proceedings of the 38th International Conference on Machine Learning, PMLR 139, 2021. Copyright 2021 by the author(s).



(a) Synthetic N^{Ω} + synthetic G

(b) Synthetic N^{Ω} + real G (Traud et al., 2012)

Figure 1. Performance comparison of various algorithms for latent preference estimation with graph side information. The x-axis is the probability of observing each rating (p), and the y-axis is the estimation error measured in $\|\cdot\|_{\max}$ or the mean absolute error (MAE). (a) Our algorithm vs (Ahn et al., 2018) where $d = 2, p_1 = 0.3, p_2 = 0.62$. (Ahn et al., 2018) performs badly due to the asymmetry of latent preference levels. (b) Our algorithm vs various algorithms proposed in the literature on real graph data and synthetic ratings. Observe that ours strictly outperforms all the existing algorithms for almost all tested values of p.

Table 1. MAE comparison with other algorithms on real N^{Ω} + real G (Massa & Avesani, 2007; Massa et al., 2008)

ItemAvg	UserAvg	User k-NN	Item k-NN	BiasedMF	SocialMF	SoRec	SoReg	TrustSVD	Ahn's	Ours
0.547	0.731	0.614	0.664	0.592	0.591	0.592	0.576	0.567	0.567	**0.540**

our knowledge, this work is the first theoretical work that shows the optimal sample complexity of latent preference estimation with graph side information without making strict assumptions on the rating generation model, made in all the prior work (Ahn et al., 2018; Yoon et al., 2018; Elmahdy et al., 2020; Zhang et al., 2021). We also develop an algorithm with low computational complexity, and our algorithm is shown to consistently outperform all the proposed algorithms in the literature including those of (Ahn et al., 2018) on synthetic/real data.

To further highlight the limitation of the proposed algorithms developed under the strict assumptions used in the literature, we present various experimental results in Fig. 1. (We will revisit the experimental setting in Sec. 6.) In Fig. 1a, we compare our algorithm with that of (Ahn et al., 2018) on synthetic rating N^{Ω} and synthetic graph G. Here, we set $d = 2, p_1 = 0.3, p_2 = 0.62$, i.e., the symmetry assumption $p_1 + p_2 = 1$ does not hold anymore. We can see that our algorithm significantly outperforms the algorithm of (Ahn et al., 2018) in terms of the estimation error for all tested values of p, where p denotes the probability of observing each rating. This clearly shows that their algorithm quickly breaks down even when the modeling assumption is just slightly off. Shown in Fig. 1b is the performance of various algorithms on synthetic rating/real graph, and we observe that the estimation error of (Ahn et al., 2018) increases as the observation rate p increases unlike all the other algorithms. (We discuss why this unexpected phenomenon happens in more details in Sec. 6.) On the other hand, our algorithm outperforms all the existing baseline algorithms for almost all tested values of p and does not exhibit any unexpected phenomenon. In Table 1, we observe that our algorithm outperforms all the other algorithms even on real rating/real graph data, although the improvement is not significant than the one for synthetic rating/real graph data. These results demonstrate the practicality of our new algorithm, which is developed under a more realistic model without limiting assumptions.

This paper is organized as follows. Related works are given in Sec. 2. We propose a generalized problem formulation for a recommender system with social graph information in Sec. 3. Sec. 4 characterizes the optimal sample complexity with main theorems. In Sec. 5, we propose an algorithm with low time complexity and provide a theoretical performance guarantee. In Sec. 6, experiments are conducted on synthetic and real data to compare the performance between our algorithm and existing algorithms in the literature. Finally, we discuss our results in Sec. 7. All the proofs and experimental details are given in the appendix.

1.1. Notation

Let $[n] = \{1, 2, \ldots, n\}$ where n is a positive integer, and let $\mathbb{1}(\cdot)$ be the indicator function. An undirected graph G

is a pair (V, E) where V is a set of vertices and E is a set of edges. For two subsets X and Y of the vertex set V, $e(X, Y)$ denotes the number of edges between X and Y.

2. Related Work

Collaborative filtering has been widely used to design recommender systems. There are two types of methods commonly used in collaborative filtering; neighborhood-based method and matrix factorization-based method. The neighborhood-based approach predicts users' ratings by finding similarity between users (Herlocker et al., 1999), or by finding similarity between items (Sarwar et al., 2001; Linden et al., 2003). In the matrix factorization-based approach, it assumes users' latent preference matrix is of a certain structure, e.g., low rank, so the latent preference matrix can be decomposed into two matrices of low dimension (Rennie & Srebro, 2005; Salakhutdinov & Mnih, 2007; 2008; Agarwal & Chen, 2010). In particular, Davenport et al. (2014) consider binary (1-bit) matrix completion and show that the maximum likelihood estimate is accurate under suitable conditions.

Since collaborative filtering relies solely on past interactions between users and items, it suffers from the cold start problem; collaborative filtering cannot provide a recommendation for new users since the system does not have enough information. A lot of works have been done to resolve this issue by incorporating social graph information into recommender systems. In specific, the social graph helps neighborhood-based method to find better neighborhood (Jamali & Ester, 2009a;b; Yang et al., 2012; 2013b). Some works add social regularization terms to the matrix factorization method to improve the performance (Cai et al., 2011; Jamali & Ester, 2010; Ma et al., 2011; Kalofolias et al., 2014).

Few works have been conducted to provide theoretical guarantees of their models that consider graph side information. Chiang et al. (2015) consider a model that incorporates general side information into matrix completion, and provide statistical guarantees. Rao et al. (2015) derive consistency guarantees for graph regularized matrix completion.

Recently, several works have studied the binary rating estimation problem with the aid of social graph information (Ahn et al., 2018; Yoon et al., 2018; Zhang et al., 2020; Elmahdy et al., 2020; Zhang et al., 2021). These works characterize the optimal sample complexity as the minimum number of observed ratings for reliable recovery of a latent preference matrix under various settings, and find how much the social graph information reduces the optimal sample complexity. In specific, Ahn et al. (2018) study the case where users are clustered in two equal-sized groups, and Yoon et al. (2018) generalize the results of (Ahn et al., 2018) to the multi-cluster case. Zhang et al. (2020; 2021) study the problem where both user-to-user and item-to-item similarity graphs are available. Lastly, Elmahdy et al. (2020) adopt the hierarchical stochastic block model to handle the case where each cluster can be grouped into sub-clusters. However, all of these works require strict assumptions on the rating generation model, which is too limited to well capture the real-world data.

Our problem can also be viewed as "node label inference on SBM," where nodes are users, edges are for social connections, node labels are m-dimensional rating vectors (consisting of $-1, 0, 1$), and node label distributions are determined by the latent preference matrix. Various works have studied recovery of clusters in SBM in the presence of node labels (Yang et al., 2013a; Saad & Nosratinia, 2018) or edge labels (Heimlicher et al., 2012; Jog & Loh, 2015; Yun & Proutiere, 2016). While their goal is recovery of clusters, Xu et al. (2014) study "edge label inference on SBM" whose goal is to recover edge label distributions as well as clusters.

Remark 1. While our problem shares high similarities with "edge label" inference on SBM, studied in (Xu et al., 2014), there exist some critical differences. To see the difference, consider a very sparse graph where many nodes are isolated. Edge label inference is impossible in this regime since there is no observed information about those isolated nodes (see Thm. 2 in (Xu et al., 2014) for more details). On the other hand, in node labelled cases, we still observe information about isolated nodes from their node labels, so it is possible to infer node label distributions as long as we observe enough number of node labels.

3. Problem Formulation

Let $[n]$ be the set of users, and let $[m]$ be the set of items where m can scale with n. For $i \in [n]$ and $j \in [m]$, R_{ij} denotes user i's *latent preference level* on item j, that is, user i's rating on item j is $+1$ (like) with probability R_{ij} or -1 (dislike) with probability $1-R_{ij}$. We assume that latent preference levels take values in the discrete set $\{p_1, p_2, \ldots, p_d\}$ where $d \geq 1$ and $0 < p_1 < \cdots < p_d < 1$. The *latent preference matrix* R is the $n \times m$ matrix whose (i, j)-th entry is R_{ij}. The *latent preference vector* of user i is the i-th row of R.

We further assume that n users are clustered into K clusters, and the users in the same cluster have the same latent preference vector. More precisely, let $C : [n] \to [K]$ be the cluster assignment function where $C(i) = k$ if user i belongs to the k-th cluster. The inverse image $C^{-1}(\{k\})$ is the set of users whose cluster assignment is k, so the users in $C^{-1}(\{k\})$ have the same latent preference vector by the assumption. We denote the latent preference vector of the users in $C^{-1}(\{k\})$ by u_k for $k \in [K]$. Note that the latent preference matrix R can be completely recovered with the cluster assignment function $C : [n] \to [K]$ and the corresponding preference vectors $u_1, \dots, u_K$.

As the latent preference vector and the cluster assignment function are generally unknown in the real world, we estimate them with observed ratings on items and the social graph.

Observed rating matrix N^{Ω} We assume that we observe binary ratings of users independently with probability p where $p \in [0, 1]$. We denote a set of observed entries by Ω which is a subset of $[n] \times [m]$. Then, the (i, j)-th entry of the observed rating matrix N^{Ω} is defined by user i's rating on item j if $(i, j) \in \Omega$ and 0 otherwise. That is, $(N^{\Omega})_{ij} \overset{\text{iid}}{\sim} \text{Bern}(p) \cdot (2\text{Bern}(R_{ij}) - 1)$.

Observed social graph G We observe the social graph $G = ([n], E)$ on n users, and we further assume that the graph is generated as per the stochastic block model (SBM) (Holland et al., 1983). Specifically, we consider the symmetric SBM. If two users i and j are from the same cluster, an edge between them is placed with probability α, independently of the others. If they are from the different clusters, the probability of having an edge between them is β, where $\alpha \geq \beta$.

Fig. 2 provides a toy example that visualizes how our observation model is realized by the latent preference matrix and the cluster assignment. Given this observation model, the goal of latent preference estimation with graph side information is to find an estimator $\psi(N^{\Omega}, G)$ that estimates the latent preference matrix R.

Remark 2 (Why binary rating?)**.** Binary rating has its critical applications such as click/impression-based advertisement recommendation, in which only -1 (shown, not clicked), 0 (not shown), 1 (shown, clicked) information is available. Moreover, binary rating is gaining increasing interests in the industry due to its simplicity and robustness. This is precisely why Youtube and Netflix, two of the largest media recommendation systems, have discarded their "star rating systems" and employed binary ratings in 2009 (Gruber, 2017) and in 2017 (Center, 2017), respectively.

Remark 3. Ahn et al. (2018) assume that each user rates each item either $+1$ (like) or -1 (dislike), and that the observations are noisy so that they can be flipped with probability $\theta \in (0, \frac{1}{2})$. This assumption can be interpreted as each user rates each item $+1$ with probability $1 - \theta$ (when the user's true rating is $+1$) or θ (when the user's true rating is -1). Therefore, our model reduces to the model of (Ahn et al., 2018) by setting $d = 2, p_1 = \theta, p_2 = 1 - \theta, K = 2, |C^{-1}(\{1\})| = |C^{-1}(\{2\})| = \frac{n}{2}$. As mentioned in Sec. 1, the parametric model used in (Ahn et al., 2018) is very limited. For example, consider the following two latent preference matrices $R_1 = \begin{bmatrix} 1/4 & 1/4 & 3/4 & 3/4 \\ 1/4 & 1/4 & 1/4 & 1/4 \end{bmatrix}$, $R_2 = \begin{bmatrix} 1/3 & 1/4 & 3/4 & 3/4 \\ 1/3 & 1/4 & 1/4 & 1/4 \end{bmatrix}$ where $n = 2, m = 4$. Then R_1 can be represented by the model used in [Ahn et al., 2018] with $\theta = \frac{1}{4}$, but R_2 cannot be handled by their model with any choice of θ since there are more than two latent preference levels in R_2.

Remark 4. Without graph observation, our observation model reduces to a special case of the observation model for the binary (1-bit) matrix completion shown in Sec. 2.1. of (Davenport et al., 2014).

4. Fundamental Limit on Sample Complexity

We now characterize the fundamental limit on the sample complexity. We first focus on the two equal-sized clusters case (i.e., $K = 2, |C^{-1}(\{1\})| = |C^{-1}(\{2\})| = \frac{n}{2}$) and will extend the results to the multi-cluster case. We use A_R and B_R for the ground-truth clusters and u_R and v_R for the corresponding latent preference vectors, respectively. We define the worst-case error probability as follows.

Definition 1 (Worst-case probability of error for two equal-sized clusters)**.** *Let γ be a fixed number in $(0, 1)$ and ψ be an estimator that outputs a latent preference matrix in $\{p_1, p_2, \dots, p_d\}^{n \times m}$ based on N^{Ω} and G. We define the worst-case probability of error $P_e^{\gamma}(\psi) := \max\big\{ \Pr(\psi(N^{\Omega}, G) \neq R) : R \in \{p_1, p_2, \dots, p_d\}^{n \times m}, \|u_R - v_R\|_0 = \lceil \gamma m \rceil \big\}$ where $\| \cdot \|_0$ is the hamming distance.*

A latent preference level $p_i \in [p_1, \dots, p_d]$ implies that the probability of choosing $(+1, -1)$ is $(p_i, 1 - p_i)$ respectively, so it corresponds to a discrete probability distribution $(p_i, 1 - p_i)$. For two latent preference levels $p_i, p_j \in [p_1, \dots, p_d]$, *the Hellinger distance* between two discrete probability distributions $(p_i, 1 - p_i)$ and $(p_j, 1 - p_j)$, denoted $d_H(p_i, p_j)$, is

$$\frac{1}{\sqrt{2}} \sqrt{(\sqrt{p_i} - \sqrt{p_j})^2 + (\sqrt{1 - p_i} - \sqrt{1 - p_j})^2}.$$

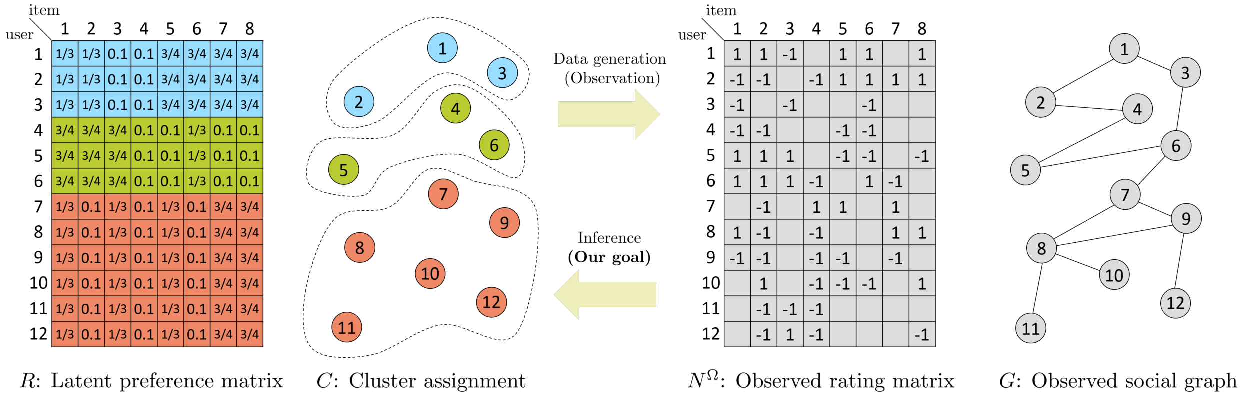
Then, *the minimum Hellinger distance* of the set of discrete-valued latent preference levels $\{p_1, \dots, p_d\}$, denoted $d_H^{\min}$, is

$$\min\{d_H(p_i, p_j) : i \neq j \in [d]\}.$$

Below is our main theorem that characterizes a sharp threshold of p, the probability of observing each rating of users, for reliable recovery as a function of $n, m, \gamma, \alpha, \beta, d_H^{\min}$.

Theorem 1. *Let $K = 2, |C^{-1}(\{1\})| = |C^{-1}(\{2\})| = \frac{n}{2}, \gamma \in (0, 1)$, $m = \omega(\log n)$, $\log m = o(n)$, I_s[1] $:= -2 \log \big(1 - d_H^2(\alpha, \beta)\big)$. Then, the following holds for arbitrary $\epsilon > 0$.*

[1] Ahn et al. (2018) made implicit assumptions that $\alpha, \beta \to 0$



user \ item	1	2	3	4	5	6	7	8
1	1/3	1/3	0.1	0.1	3/4	3/4	3/4	3/4
2	1/3	1/3	0.1	0.1	3/4	3/4	3/4	3/4
3	1/3	1/3	0.1	0.1	3/4	3/4	3/4	3/4
4	3/4	3/4	3/4	0.1	0.1	1/3	0.1	0.1
5	3/4	3/4	3/4	0.1	0.1	1/3	0.1	0.1
6	3/4	3/4	3/4	0.1	0.1	1/3	0.1	0.1
7	1/3	0.1	1/3	0.1	1/3	0.1	3/4	3/4
8	1/3	0.1	1/3	0.1	1/3	0.1	3/4	3/4
9	1/3	0.1	1/3	0.1	1/3	0.1	3/4	3/4
10	1/3	0.1	1/3	0.1	1/3	0.1	3/4	3/4
11	1/3	0.1	1/3	0.1	1/3	0.1	3/4	3/4
12	1/3	0.1	1/3	0.1	1/3	0.1	3/4	3/4

R: Latent preference matrix

C: Cluster assignment

user \ item	1	2	3	4	5	6	7	8
1	1	1	-1		1	1		1
2	-1	-1		-1	1	1	1	1
3	-1		-1			-1		
4	-1	-1			-1	-1		
5	1	1	1		-1	-1		-1
6	1	1	1	-1		1	-1	
7		-1		1	1		1	
8	1	-1		-1			1	1
9	-1	-1		-1	-1		-1	
10		1		-1	-1	-1		1
11		-1	-1	-1				
12		-1	1	-1				-1

N^{Ω}: Observed rating matrix

G: Observed social graph

Figure 2. A toy example of our model where $n = 12, m = 8, d = 3, p_1 = 0.1, p_2 = \frac{1}{3}, p_3 = \frac{3}{4}, K = 3, C^{-1}(\{1\}) = \{1,2,3\}, C^{-1}(\{2\}) = \{4,5,6\}, C^{-1}(\{3\}) = \{7,8,9,10,11,12\}, p = 0.5, \alpha = 0.6, \beta = 0.1$.

(I) if $p \geq \frac{1}{(d_H^{\min})^2} \max\left\{\frac{(1+\epsilon)\log n - \frac{n}{2}I_s}{\gamma m}, \frac{(1+\epsilon)2\log m}{n}\right\}$*, then there exists an estimator* ψ *that outputs a latent preference matrix in* $\{p_1, p_2, \ldots, p_d\}^{n\times m}$ *based on* N^{Ω} *and* G *such that* $P_e^{\gamma}(\psi) \to 0$ *as* $n \to \infty$.

(II) if $p \leq \frac{1}{(d_H^{\min})^2} \max\left\{\frac{(1-\epsilon)\log n - \frac{n}{2}I_s}{\gamma m}, \frac{(1-\epsilon)2\log m}{n}\right\}$ *and* $\alpha = O(\frac{\log n}{n})$*, then* $P_e^{\gamma}(\psi) \nrightarrow 0$ *as* $n \to \infty$ *for any* ψ.

Remark 5. We note that our technical contributions lie in the proof of Thm. 1. In specific, we find the upper bound of the probability of error in Lem. 3 by using the results of Lem. 1, 2, and we made nontrivial technical contributions as we need to handle a significantly larger set of candidate latent preference matrices.

Theorem 1 shows that $\frac{1}{(d_H^{\min})^2} \max\left\{\frac{\log n - \frac{n}{2}I_s}{\gamma m}, \frac{2\log m}{n}\right\}$ can be used as a sharp threshold for reliable recovery of the latent preference matrix. As nmp is the expected number of observed entries, we define the optimal sample complexity for two-cluster cases as follows.

Definition 2. $p^*_{(\gamma)} := \frac{1}{(d_H^{\min})^2} \max\left\{\frac{\log n - \frac{n}{2}I_s}{\gamma m}, \frac{2\log m}{n}\right\}$ *denotes the optimal observation rate. Then* $nmp^*_{(\gamma)} = \frac{1}{(d_H^{\min})^2} \max\left\{\frac{1}{\gamma}(n\log n - \frac{1}{2}n^2 I_s), 2m\log m\right\}$ *denotes the optimal sample complexity for two-cluster cases.*

The optimal sample complexity for two-cluster cases is written as a function of $p_1, ..., p_d$, so the dependency on d is implicit. To see the dependency clearly, we can set $p_i = \frac{i}{d+1}$. This gives us $p^*_{(\gamma)} \approx 2d^2 \max\left\{\frac{\log n - \frac{n}{2}I_s}{\gamma m}, \frac{2\log m}{n}\right\}$, and $p^*_{(\gamma)}$ increases as a quadratic function of d.

Remark 6 (How does the graph information reduce the optimal sample complexity?)**.** One can observe that I_s decreases as α and β get closer to each other, and $I_s = 0$ when $\alpha = \beta$. Hence I_s measures the quality of the graph information. If we consider the case that does not employ the graph information, it is equivalent to the case of $\alpha = \beta$ ($I_s = 0$) in our model, thereby getting the optimal sample complexity of $\frac{1}{(d_H^{\min})^2} \max\left\{\frac{1}{\gamma}n\log n, 2m\log m\right\}$. Therefore, exploiting the graph information results in the reduction of the optimal sample complexity by $\frac{1}{(d_H^{\min})^2}\frac{1}{2\gamma}n^2 I_s$ provided that $\frac{1}{\gamma}n\log n > 2m\log m$. Note that the optimal sample complexity stops decreasing when I_s is larger than a certain threshold which implies the gain is saturated.

Remark 7. If we set $d = 2, p_1 = \theta, p_2 = 1-\theta$, then $(d_H^{\min})^2 = 1 - 2\sqrt{\theta(1-\theta)} = (\sqrt{1-\theta} - \sqrt{\theta})^2$. Plugging this into the result of Theorem 1, we get $p^*_{(\gamma)} = \frac{1}{(\sqrt{1-\theta}-\sqrt{\theta})^2} \max\left\{\frac{\log n - \frac{n}{2}I_s}{\gamma m}, \frac{2\log m}{n}\right\}$, recovering the main theorem of (Ahn et al., 2018) as a special case of our result.

Our results can be extended to the case of multiple (possibly unequal-sized) clusters by combining the technique developed in Theorem 1 and the technique of (Yoon et al., 2018). Suppose $d_H(p_i, p_j)$ achieves the minimum Hellinger distance when $p_i = p_{d_0}, p_j = p_{d_0+1}$. Define $\mathbb{p} : \{p_1, \ldots, p_d\}^m \to \{p_{d_0}, p_{d_0+1}\}^m$ that maps a latent preference vector to a latent preference vector consisting of latent preference levels $\{p_{d_0}, p_{d_0+1}\}$. In explicit, $\mathbb{p}$ sends each coordinate x_i of a latent preference vector to p_{d_0} if $x_i \leq p_{d_0}$; p_{d_0+1} if $x_i \geq p_{d_0+1}$. We now present the extended result below, while deferring the the proof to the appendix.

Theorem 2. *Let* $m = \omega(\log n)$, $\log m = o(n)$, $c_k =$

and $\frac{\alpha}{\beta} \to 1$ as $n \to \infty$. These assumptions are used when they approximate $-2\log(1 - d_H^2(\alpha,\beta)) = (1+o(1))(\sqrt{\alpha} - \sqrt{\beta})^2$. The approximation does not hold without above assumptions, in explicit, $-2\log(1 - d_H^2(\alpha,\beta)) = (\sqrt{\alpha} - \sqrt{\beta})^2\left\{\frac{(\sqrt{\alpha}+\sqrt{\beta})^2}{4\beta(1-\beta)} + o(1)\right\}$ (see the appendix for the derivation). The MLE achievability part of our theorem does not make any implicit assumptions, and the results hold for any α and β with our modified definition of $I_s := -2\log(1 - d_H^2(\alpha,\beta))$.

$|C^{-1}(\{k\})|, c_{i,j} = \frac{c_i+c_j}{2}$, $\liminf_{n\to\infty} \frac{c_k}{n} > 0$ *for all* $k \in [K]$, $\liminf_{m\to\infty} \frac{\|\mathbb{p}(u_i)-\mathbb{p}(u_j)\|_0}{m} > 0$ *for all* $i \neq j \in [K]$. *Then, the following holds for arbitrary* $\epsilon > 0$.

(I) (achievability) If $p \geq \frac{1}{(d_H^{\min})^2} \max\left\{ \max_{i\neq j\in[K]} \left\{ \frac{(1+\epsilon)\log n - c_{i,j} I_s}{\|\mathbb{p}(u_i)-\mathbb{p}(u_j)\|_0} \right\}, \max_{k\in[K]} \left\{ \frac{(1+\epsilon)\log m}{c_k} \right\} \right\}$ *then there exists an estimator* ψ *such that* $\Pr(\psi(N^{\Omega}, G) \neq R) \to 0$ *as* $n \to \infty$.

(II) (impossibility)
Suppose $R \in \{p_{d_0}, p_{d_0+1}\}^{n\times m}$, $\alpha = O(\frac{\log n}{n})$. *If* $p \leq \frac{1}{(d_H^{\min})^2} \max\left\{ \max_{i\neq j\in[K]} \left\{ \frac{(1-\epsilon)\log n - c_{i,j} I_s}{\|\mathbb{p}(u_i)-\mathbb{p}(u_j)\|_0} \right\}, \max_{k\in[K]} \left\{ \frac{(1-\epsilon)\log m}{c_k} \right\} \right\}$ *then* $\Pr(\psi(N^{\Omega}, G) \neq R) \nrightarrow 0$ *as* $n \to \infty$ *for any* ψ.

Remark 8. One can observe that Theorem 1 is a special case of Theorem 2 by setting $K = 2, c_1 = c_2 = \frac{n}{2}, \|\mathbb{p}(u_1) - \mathbb{p}(u_2)\|_0 = \lceil \gamma m \rceil$.

Remark 9. In light of Theorem 14 in (Abbe, 2018), we conjecture that our results can be extended to asymmetric SBMs with a new definition of I_s involving Chernoff-Hellinger divergence.

5. Our Proposed Algorithm

In this section, we develop a computationally efficient algorithm that can recover the latent preference matrix R without knowing the latent preference levels $\{p_1, \ldots, p_d\}$. We then provide a theoretical guarantee that if $p \geq \frac{1}{(d_H^{\min})^2} \max\left\{ \frac{(1+\epsilon)\log n - \frac{n}{2} I_s}{\gamma m}, \frac{(1+\epsilon) 2\log m}{n} \right\}$ for some $\epsilon > 0$, then the proposed algorithm recovers the latent preference matrix with high probability. Now we provide a high-level description of our algorithm while deferring the pseudocode to the appendix.

Algorithm description

Input: $N^{\Omega} \in \{-1, 0, +1\}^{n\times m}, G = ([n], E), K, d, \ell_{\max}$

Output: Clusters of users $A_1^{(\ell_{\max})}, \ldots, A_K^{(\ell_{\max})}$, latent preference vectors $\hat{u_1}^{(\ell_{\max})}, \ldots, \hat{u_K}^{(\ell_{\max})}$

Stage 1. Partial recovery of clusters We run a spectral method (Gao et al., 2017) on G to get an initial clustering result $A_1^{(0)}, \ldots, A_K^{(0)}$. Unless α is too close to β, this stage will give us a reasonable clustering result, with which we can kick-start the entire estimation procedure. Other clustering algorithms (Abbe & Sandon, 2015; Chin et al., 2015; Krzakala et al., 2013; Lei & Rinaldo, 2015) can also be used for this stage.

Stage 2 We iterate **Stage 2-(i)** and **Stage 2-(ii)** for $\ell = 1, \ldots, \ell_{\max}$.

Stage 2-(i). Recovery of latent preference vectors In the ℓ-th iteration step, this stage takes the clustering result $A_1^{(\ell-1)}, \ldots, A_K^{(\ell-1)}$ and rating data N^{Ω} as input and outputs the estimation of latent preference vectors $\hat{u_1}^{(\ell)}, \ldots, \hat{u_K}^{(\ell)}$.

First, for each cluster $A_k^{(\ell-1)}$, we estimate the latent preference levels for $d\lceil \log m \rceil$ randomly chosen items with replacement. The estimation of a latent preference level can be easily done by computing the ratio of "the number of $+1$ ratings" to "the number of observed ratings (i.e., nonzero ratings)" for each item within the cluster $A_k^{(\ell-1)}$. Now we have $Kd\lceil \log m \rceil$ number of estimations, and these estimations will be highly concentrated around the latent preference levels $p_1, \ldots, p_d$ under our modeling assumptions (see the appendix for the mathematical justifications). After running a distance-based clustering algorithm (see the pseudocode for details), we take the average within each cluster to get the estimations $\hat{p_1}^{(\ell)}, \ldots, \hat{p_d}^{(\ell)}$.

Given the estimations $\hat{p_1}^{(\ell)}, \ldots, \hat{p_d}^{(\ell)}$ and the clustering result $A_1^{(\ell-1)}, \ldots, A_K^{(\ell-1)}$, we estimate latent preference vectors $\hat{u_1}^{(\ell)}, \ldots, \hat{u_K}^{(\ell)}$ by maximizing the likelihood of the observed rating matrix N^{Ω} and the observed social graph $G = ([n], E)$. In specific, the j-th coordinate of $\hat{u_k}^{(\ell)}$ can be obtained by finding $\arg\min_{\hat{p_h}^{(\ell)}: h\in[d]} \hat{L}(\hat{p_h}^{(\ell)}; A_k^{(\ell-1)}, j)$ where

$$\hat{L}(\hat{p_h}^{(\ell)}; A_k^{(\ell-1)}, j) := \sum_{i\in A_k^{(\ell-1)}} \big\{ \mathbb{1}(N_{ij}^{\Omega} = 1)(-\log \hat{p_h}^{(\ell)}) + \mathbb{1}(N_{ij}^{\Omega} = -1)(-\log(1 - \hat{p_h}^{(\ell)})) \big\}.$$

Stage 2-(ii). Refinement of clusters In the ℓ-th iteration step, this stage takes the clustering result $A_1^{(\ell-1)}, \ldots, A_K^{(\ell-1)}$, the estimation of latent preference vectors $\hat{u_1}^{(\ell)}, \ldots, \hat{u_K}^{(\ell)}$, rating data N^{Ω}, graph data G as input and outputs the refined clustering result $A_1^{(\ell)}, \ldots, A_K^{(\ell)}$.

We first compute $\hat{\alpha}, \hat{\beta}$ that estimate α, β based on the clustering result $A_1^{(\ell-1)}, \ldots, A_K^{(\ell-1)}$ and the number of edges within a cluster and across clusters. Let $A_k^{(\ell-1,0)} := A_k^{(\ell-1)}$ for $k \in [K]$. Then $A_k^{(\ell-1,0)}$'s are iteratively refined by $T = \lceil \log_2 n \rceil$ times of refinement steps as follows.

Suppose we have a clustering result $A_k^{(\ell-1,t-1)}$'s from the $(t-1)$-th refinement step where $t = 1, \ldots, T$. Given the estimations $\hat{\alpha}, \hat{\beta}$, the estimated latent preference vectors $\hat{u_1}^{(\ell)}, \ldots, \hat{u_K}^{(\ell)}$, and the clustering result $A_1^{(\ell-1,t-1)}, \ldots, A_K^{(\ell-1,t-1)}$, we find the refined clustering result $A_1^{(\ell-1,t)}, \ldots, A_K^{(\ell-1,t)}$ by updating each user's affiliation. Specifically, for each user i, we put user i to $A_{k^*}^{(\ell-1,t)}$ where $k^* := \operatorname{argmin}_{k\in[K]} \hat{L}(A_k^{(\ell-1,t-1)}; i)$ and $\hat{L}(A_k^{(\ell-1,t-1)}; i) := -\log(\hat{\alpha}) e(\{i\}, A_k^{(t-1)}) - \log(1 - \hat{\alpha})\{|A_k^{(\ell-1,t-1)}| -$

$e(\{i\}, A_k^{(\ell-1,t-1)})\} + \sum_{k' \neq k} \Big\{ -\log(\hat{\beta}) e(\{i\}, A_{k'}^{(\ell-1,t-1)}) - \log(1 - \hat{\beta})\{|A_{k'}^{(\ell-1,t-1)}| - e(\{i\}, A_{k'}^{(\ell-1,t-1)})\}\Big\} + \Big\{ - \sum_{j:N_{ij}^{\Omega}=1} \log(\hat{u_k}^{(\ell)})_j - \sum_{j:N_{ij}^{\Omega}=-1} \log(1 - (\hat{u_k}^{(\ell)})_j)\Big\}$.

($(\hat{u_k}^{(\ell)})_j$ denotes the j-th coordinate of $\hat{u_k}^{(\ell)}$.) In each refinement step, the number of mis-clustered users will decrease provided that estimations $\hat{u_k}^{(\ell)}$'s, $\hat{\alpha}, \hat{\beta}$ are close enough to their true values (see the appendix for the mathematical justifications).

After T times of refinement steps, we let $A_k^{(\ell)} := A_k^{(\ell-1,T)}$ for $k \in [K]$. Finally, this stage outputs the refined clustering result $A_1^{(\ell)}, \dots, A_K^{(\ell)}$.

Remark 10. The computational complexity of our algorithm can be computed as follows; $O(|E| \log n)$ for **Stage 1** via the power method (Boutsidis et al., 2015), $O(|\Omega|)$ for **Stage 2-(i)**, $O((|\Omega| + |E|) \log n)$ for **Stage 2-(ii)**. As ℓ_{max} is constant, the linear factor of ℓ_{max} is omitted in the computational complexity of **Stage 2-(i),(ii)**. Overall, our algorithm has low computational complexity of $O((|\Omega| + |E|) \log n)$.

Remark 11. We note that our technical contributions lie in the analysis of **Stage 2-(i)** while the analysis of **Stage 1** and **Stage 2-(ii)** is similar to those in (Ahn et al., 2018; Yoon et al., 2018). In specific, we sample $O(\lceil \log m \rceil)$ number of items in **Stage 2-(i)** to get estimations of the latent preference levels and Lem. 8 ensures that those estimations are located in the $o(1)$-radius neighborhoods of ground-truth latent preference levels with high probability. Then Lem. 9 ensures that estimations of latent preference vectors converges to ground-truth latent preference vectors with high probability.

For the two equal-sized clusters case, the following theorem asserts that our algorithm will successfully estimate the latent preference matrix with high probability as long as the sampling probability is slightly above the optimal threshold. We defer the proof to the appendix.

Theorem 3. *Let $\ell_{\max} = 1, K = 2, |C^{-1}(\{1\})| = |C^{-1}(\{2\})| = \frac{n}{2}, \gamma \in (0,1)$, $m = \omega(\log n)$, $\log m = o(n)$, $(\sqrt{\alpha} - \sqrt{\beta})^2 = \omega(\frac{1}{n})$, $m = O(n)$, and $\alpha = O(\frac{\log n}{n})$. Let ϕ_j be the ratio of the number of p_j's among $(u_R)_1, \dots, (u_R)_m, (v_R)_1, \dots, (v_R)_m$ to $2m$ for $j = 1, \dots, d$, and assume that $\phi_j \nrightarrow 0$ as $n \to \infty$. If*

$$p \geq \frac{1}{(d_H^{\min})^2} \max\left\{ \frac{(1+\epsilon)\log n - \frac{n}{2} I_s}{\gamma m}, \frac{2(1+\epsilon)\log m}{n} \right\}$$

for some $\epsilon > 0$, then our algorithm outputs $\hat{R}$ where the following holds with probability approaching to 1 as n goes to ∞ : $\|\hat{R} - R\|_{\max} := \max_{(i,j) \in [n]\times[m]} |\hat{R}_{ij} - R_{ij}| = o(1)$.

Remark 12. As our algorithm makes use of only graph data at Stage 1, the initial clustering result highly depends on the quality of graph data I_s. In the extreme cases where only rating data are available, Stage 1 will output a meaningless clustering result. As the performance of Stage 2 depends on the success of Stage 1, our algorithm may not work well even if the observation rate p is above the optimal rate. In Sec. E, we suggest an alternative algorithm, which utilizes both rating and graph data at Stage 1. Analyzing the performance of this new algorithm is an interesting open problem.

6. Experimental Results

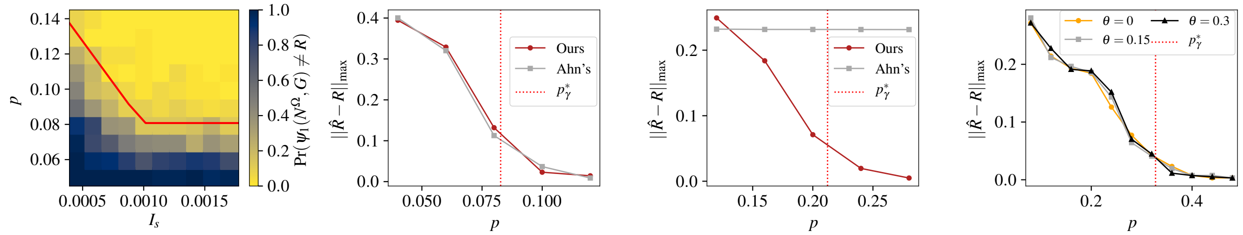
In this section, we run several experiments to evaluate the performance of our proposed algorithm. Denoting by $\hat{R}$ the output of an estimator, the estimation quality is measured by the max norm of the error matrix, i.e., $\|\hat{R} - R\|_{\max} := \max_{(i,j)\in[n]\times[m]} |\hat{R}_{ij} - R_{ij}|$. For each observation rate p, we generate synthetic data (N^{Ω}, G) 100 times at random and then report the average errors.

6.1. Non-asymptotic Performance of Our Algorithm

Shown in Fig. 3a is the probability of error $\Pr(\psi_1(N^{\Omega}, G) \neq R)$ of our algorithm for $(n, m, K, d) = (10000, 5000, 2, 3)$ and various combinations of (I_s, p). To measure $\Pr(\psi_1(N^{\Omega}, G) \neq R)$, we allow our algorithm to have access to the latent preference levels $(p_1, p_2, p_3) = (0.2, 0.5, 0.7)$ in Stage 2. We draw p_{γ}^* as a red line. While the theoretical guarantee of our algorithm is valid when n, m go to ∞, Fig. 3a shows that Theorem 1 predicts the optimal observation rate p_{γ}^* with small error for sufficiently large n, m. One can observe a sharp phase transition around p_{γ}^*.

6.2. Limitation of the Symmetric Latent Preference Levels

As described in Sec. 1, the latent preference matrix model studied in (Ahn et al., 2018) assumes that the latent preference level must be either θ or $1-\theta$ for some θ, which is fully symmetric. In this section, we show that this model cannot be applied unless the symmetry assumption perfectly holds. Let $(K, d, n, m, \gamma, \alpha, \beta) = (2, 2, 2000, 1000, \frac{1}{4}, 0.7, 0.3)$. Shown in Fig. 3b, Fig. 1a, Fig. 3c are the estimation errors of our algorithm and that of the algorithm proposed in (Ahn et al., 2018) for various pairs of (p_1, p_2). (1) Fig. 3b shows the result for $(p_1, p_2) = (0.3, 0.7)$ where the latent preference levels are perfectly symmetric, and the two algorithms perform exactly the same. (2) Fig. 1a shows the result for $(p_1, p_2) = (0.3, 0.62)$ where the latent preference levels are slightly asymmetric. The estimation error of the algorithm of (Ahn et al., 2018) is much larger than ours for all



(a) Phase transition (b) Symmetric levels (c) Asymmetric levels (d) Noisy SBM

Figure 3. (a) Non-asymptotic performance of our algorithm. One can observe a sharp phase transition around p_γ^* (b), (c) Limitation of the symmetric level model. (b) When the latent preference levels are symmetric ($p_1 = 0.3$ and $p_2 = 0.7$), our algorithm and the algorithm proposed in (Ahn et al., 2018) achieve the same estimation errors. (c) When the latent preference levels are not symmetric ($p_1 = 0.3$ and $p_2 = 0.55$), our algorithm significantly outperforms the one proposed in (Ahn et al., 2018). (d) Estimation error as a function of observation rate p when graph data is generated as per noisy stochastic block models. Observe that our algorithm is robust to model errors.

tested values of p. (3) Shown in Fig. 3c are the experimental results with $(p_1, p_2) = (0.3, 0.55)$. Observe that the gap between these two algorithms becomes even larger, and the algorithm of (Ahn et al., 2018) seems not able to output a reliable estimation of the latent preference matrix due to its limited modeling assumption.

6.3. Robustness to Model Errors

We show that while the theoretical guarantee of our algorithm holds only for a certain data generation model, our algorithm is indeed robust to model errors and can be applied to a wider range of data generation models. Specifically, we add noise to the stochastic block model as follows. If two users i and j are from the same cluster, we place an edge with probability $\alpha + q_{ij}$, independently of other edges, where $q_{ij} \overset{i.i.d.}{\sim} U[-\theta, \theta]$ for some constant θ. Similarly, if they are from the two different clusters, the probability of having an edge between them is $\beta + q_{ij}$. Under this *noisy* stochastic block model, we generate data and measure the estimation errors with $(K, d, p_1, p_2, p_3, n, m, \gamma, \alpha, \beta) = (2, 3, 0.2, 0.5, 0.7, 2000, 1000, \frac{1}{4}, 0.7, 0.3)$, $\theta = 0, 0.15, 0.3$. Fig. 3d shows that the performance of our algorithm is not affected by the model noise, implying the model robustness of our algorithm. The result for $\theta = 0.3$ is even more interesting since the level of noise is so large that $\alpha + q_{ij}$ can become even lower than $\beta + q_{i'j'}$ for some (i, j) and (i', j'). However, even under this extreme condition, our algorithm successfully recovers the latent preference matrix.

6.4. Real-World Data Experiments

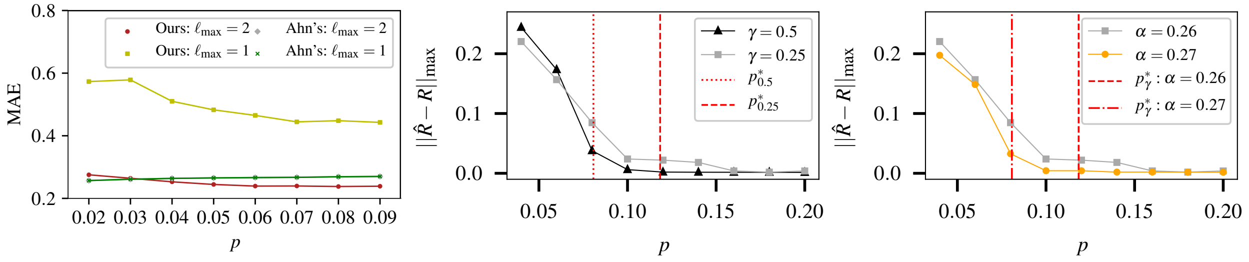
The experimental result given in Sec. 6.3 motivated us to evaluate the performance of our algorithm when real-world graph data is given as graph side information. First, we take Facebook graph data (Traud et al., 2012) as graph side information (which has a 3-cluster structure) and generate binary ratings as per our discrete-valued latent preference model ($(p_1, p_2, p_3) = (0.05, 0.5, 0.95)$). We use 80% (randomly sampled) of Ω as a training set (Ω_{tr}) and the remaining 20% of Ω as a test set (Ω_t). We use mean absolute error (MAE) $\frac{1}{|\Omega_t|} \sum_{(i,j) \in \Omega_t} \|N_{ij}^\Omega - (2\hat{R}_{ij} - 1)\|$ for the performance metric.[2] Then we compare the performance of our algorithm with other algorithms in the literature.[3] Fig. 1b shows that our algorithm outperforms other baseline algorithms for almost all tested values of p. The red dotted line is the expected value of MAE of the optimal estimator (see the appendix for a detailed explanation) which means our algorithm shows near-optimal performance. Unlike other algorithms, MAE of (Ahn et al., 2018) increases as p increases. One explanation is that the algorithm of (Ahn et al., 2018) cannot properly handle $d \geq 3$ cases due to its limited modeling assumption.

Remark 13. While our algorithm shows near-optimal performance with $\ell_{\max} = 1$ for synthetic data, Fig. 4a shows that our algorithm does not work well with $\ell_{\max} = 1$ for real-world data. This phenomenon can be explained as follows. If $\ell_{\max} = 1$, the estimations of latent preference vectors are only based on the result of the Stage 1. For real-world graph data, the clustering result of the Stage 1 may not be close to the ground-truth clusters, thereby resulting in bad estimations of latent preference vectors in Stage 2-(i). Surprisingly, our algorithm shows near-optimal performance with $\ell_{\max} = 2$ even for real-world graph data (see Fig. 1b). Unlike ours, the algorithm of (Ahn et al., 2018) shows no difference between $\ell_{\max} = 1$ and $\ell_{\max} = 2$.

Furthermore, we evaluate the performance of our algorithm

[2] We compute the difference between N_{ij}^Ω and $2\hat{R}_{ij} - 1$ for fair comparison since $N_{ij}^\Omega \in \{\pm 1\}$, $\hat{R}_{ij} \in [0, 1]$.

[3] We compare our algorithm with the algorithm of (Ahn et al., 2018), item average, user average, user k-NN (nearest neighbors), item k-NN, BiasedMF (Koren, 2008), SocialMF (Jamali & Ester, 2010), SoRec (Ma et al., 2008), SoReg (Ma et al., 2011), Trust SVD (Guo et al., 2015b). Except for ours and that of (Ahn et al., 2018), we adopt implementations from LibRec (Guo et al., 2015a).



(a) $\ell_{\max}=1$ vs $\ell_{\max}=2$ (b) $\gamma=0.5$ vs $\gamma=0.25$ (c) $\alpha=0.26$ vs $\alpha=0.27$

Figure 4. (a) Estimation error as a function of observation rate p for the Facebook graph data (Traud et al., 2012) with different values of $\ell_{\max}$. (b), (c) Estimation error as a function of observation probability p with different values of γ and α. The x-axis is the probability of observing each rating (p), and the y-axis is the estimation error measured in $\|\cdot\|_{\max}$.

on a real rating/real graph dataset called Epinions (Massa & Avesani, 2007; Massa et al., 2008). We use 5-fold cross-validation to determine hyperparameters. Then we compute MAE for a randomly sampled test set (with 500 iterations). Shown in Table 1 are MAE's for various algorithms. Although the improvement is not significant than the one for synthetic rating/real graph data, our algorithm outperforms all the other algorithms. Note that all the experimental results presented in the prior work are based on synthetic rating (Ahn et al., 2018; Yoon et al., 2018; Zhang et al., 2021), and this is the first real rating/real graph experiment that shows the practicality of binary rating estimation with graph side information.

6.5. Estimation Error with Different Values of γ and I_s

We corroborate Theorem 3. More specifically, we observe how the estimation error behaves as a function of p when γ and (α,β) varies. Let $d=3, p_1=0.2, p_2=0.5, p_3=0.7, n=10000, m=5000$. We first compare cases for $(\alpha,\beta,\gamma)=(0.26,0.23,0.5)$ and $(0.26,0.23,0.25)$. Shown in Fig. 4b is the estimation error as a function of p. We draw p^*_γ as dotted vertical lines. One can see from the figure that the estimation error for $(\alpha,\beta,\gamma)=(0.26,0.23,0.5)$ is lower than that for $(\alpha,\beta,\gamma)=(0.26,0.23,0.25)$ for all tested values of p. This can be explained by the fact that p^*_γ decreases as γ increases, as stated in Theorem 3. We also compare cases for $(\alpha,\beta,\gamma)=(0.26,0.23,0.25)$ and $(0.27,0.23,0.25)$. Note that the only difference between these cases is the value of α. By Theorem 3, we have $p^*_\gamma=0.118$ for the former case, and $p^*_\gamma=0.081$ for the latter case. That is, a larger value of α implies a higher quality of graph side information, i.e., the graph side information is more useful for predicting the latent preference matrix R. Fig. 4c shows the estimation error as a function of p, and we can see that even a small increase in the quality of the graph can result in a significant decrease in p^*_γ.

7. Conclusion

We studied the problem of estimating the latent preference matrix whose entries are discrete-valued given a partially observed binary rating matrix and graph side information. We first showed that the latent preference matrix model adopted in existing works is highly limited, and proposed a generalized data generation model. We characterized the optimal sample complexity that guarantees perfect recovery of latent preference matrix, and showed that this optimal complexity also serves as a tight lower bound, i.e., no estimation algorithm can achieve perfect recovery below the optimal sample complexity. We also proposed a computationally efficient estimation algorithm. Our analysis showed that our proposed algorithm can perfectly estimate the latent preference matrix if the sample complexity is above the optimal sample complexity. We provided experimental results that corroborate our theoretical findings, highlight the importance of our relaxed modeling assumptions, imply the robustness of our algorithm to model errors, and compare our algorithm with other algorithms on real-world data.

Acknowledgements This material is based upon work supported by NSF Award DMS-2023239.

References

Abbe, E. Community detection and stochastic block models: Recent developments. *Journal of Machine Learning Research*, 18(177):1–86, 2018.

Abbe, E. and Sandon, C. Community detection in general stochastic block models: Fundamental limits and efficient algorithms for recovery. In *Proceedings of the 2015 IEEE 56th Annual Symposium on Foundations of Computer Science*, FOCS, pp. 670–688. IEEE, 2015.

Agarwal, D. and Chen, B.-C. flda: Matrix factorization through latent dirichlet allocation. In *Proceedings of the Third ACM International Conference on Web Search and Data Mining*, pp. 91–100, 2010.

Ahn, K., Lee, K., Cha, H., and Suh, C. Binary rating estimation with graph side information. In *Advances in Neural Information Processing Systems 31*, pp. 4272–4283, 2018.

Boutsidis, C., Gittens, A., and Kambadur, P. Spectral clustering via the power method - provably. In *Proceedings of the 32nd International Conference on International Conference on Machine Learning*, pp. 40–48, 2015.

Cai, D., He, X., Han, J., and Huang, T. S. Graph regularized nonnegative matrix factorization for data representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(8):1548–1560, 2011.

Center, N. M. Goodbye stars, hello thumbs. `https://media.netflix.com/en/company-blog/goodbye-stars-hello-thumbs`, 2017.

Chiang, K.-Y., Hsieh, C.-J., and Dhillon, I. S. Matrix completion with noisy side information. In *Advances in Neural Information Processing Systems 28*, pp. 3447–3455, 2015.

Chin, P., Rao, A., and Vu, V. Stochastic block model and community detection in sparse graphs: A spectral algorithm with optimal rate of recovery. In *Proceedings of The 28th Conference on Learning Theory*, volume 40, pp. 391–423. PMLR, 2015.

Davenport, M. A., Plan, Y., van den Berg, E., and Wootters, M. 1-Bit matrix completion. *Information and Inference: A Journal of the IMA*, 3(3):189–223, 2014.

Elmahdy, A., Ahn, J., Suh, C., and Mohajer, S. Matrix completion with hierarchical graph side information. In *Advances in Neural Information Processing Systems 34*, 2020.

Gao, C., Ma, Z., Zhang, A. Y., and Zhou, H. H. Achieving optimal misclassification proportion in stochastic block models. *J. Mach. Learn. Res.*, pp. 1980–2024, 2017.

Gruber, J. Why youtube switched from 5-star ratings to thumbs up/down in 2009. `https://daringfireball.net/linked/2017/03/18/youtube-thumbs-stars`, 2017.

Guo, G., Zhang, J., Sun, Z., and Yorke-Smith, N. Librec: A java library for recommender systems. In *UMAP Workshops*, volume 1388 of *CEUR Workshop Proceedings*, 2015a.

Guo, G., Zhang, J., and Yorke-Smith, N. Trustsvd: Collaborative filtering with both the explicit and implicit influence of user trust and of item ratings. In *AAAI*, pp. 123–129, 2015b.

Heimlicher, S., Lelarge, M., and Massoulié, L. Community detection in the labelled stochastic block model. *NIPS Workshop: Algorithmic and Statistical Approaches for Large Social Networks*, 2012.

Herlocker, J. L., Konstan, J. A., Borchers, A., and Riedl, J. An algorithmic framework for performing collaborative filtering. In *Proceedings of the 22Nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 230–237, 1999.

Holland, P. W., Laskey, K. B., and Leinhardt, S. Stochastic blockmodels: First steps. *Social networks*, 5(2):109–137, 1983.

Jamali, M. and Ester, M. Trustwalker: A random walk model for combining trust-based and item-based recommendation. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 397–406, 2009a.

Jamali, M. and Ester, M. Using a trust network to improve top-n recommendation. In *Proceedings of the Third ACM Conference on Recommender Systems*, pp. 181–188, 2009b.

Jamali, M. and Ester, M. A matrix factorization technique with trust propagation for recommendation in social networks. In *Proceedings of the Fourth ACM Conference on Recommender Systems*, pp. 135–142, 2010.

Jog, V. and Loh, P.-L. Recovering communities in weighted stochastic block models. In *53rd Annual Allerton Conference on Communication, Control, and Computing*, pp. 1308–1315, 2015.

Kalofolias, V., Bresson, X., Bronstein, M., and Vandergheynst, P. Matrix completion on graphs. *NIPS Workshop, Out of the Box: Robustness in High Dimension*, 2014.

Koren, Y. Factorization meets the neighborhood: A multifaceted collaborative filtering model. In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 426–434, 2008.

Krzakala, F., Moore, C., Mossel, E., Neeman, J., Sly, A., Zdeborová, L., and Zhang, P. Spectral redemption in clustering sparse networks. *Proceedings of the National Academy of Sciences*, 110(52):20935–20940, 2013.

Lei, J. and Rinaldo, A. Consistency of spectral clustering in sparse stochastic block models. *The Annals of Statistics*, 43(1):215–237, 2015.

Linden, G., Smith, B., and York, J. Amazon.com recommendations: Item-to-item collaborative filtering. *IEEE Internet Computing 7*, pp. 76–80, 2003.

Ma, H., Yang, H., Lyu, M. R., and King, I. Sorec: Social recommendation using probabilistic matrix factorization. In *Proceedings of the 17th ACM Conference on Information and Knowledge Management*, pp. 931–940, 2008.

Ma, H., Zhou, D., Liu, C., Lyu, M. R., and King, I. Recommender systems with social regularization. In *Proceedings of the Fourth ACM International Conference on Web Search and Data Mining*, pp. 287–296, 2011.

Massa, P. and Avesani, P. Trust-aware recommender systems. In *Proceedings of the 2007 ACM Conference on Recommender Systems*, pp. 17–24, 2007.

Massa, P., Souren, K., Salvetti, M., and Tomasoni, D. Trustlet, open research on trust metrics. In *BIS*, 2008.

Rao, N., Yu, H.-F., Ravikumar, P. K., and Dhillon, I. S. Collaborative filtering with graph information: Consistency and scalable methods. In *Advances in Neural Information Processing Systems 28*, pp. 2107–2115, 2015.

Rennie, J. D. M. and Srebro, N. Fast maximum margin matrix factorization for collaborative prediction. In *Proceedings of the 22nd International Conference on Machine Learning*, pp. 713–719, 2005.

Saad, H. and Nosratinia, A. Community detection with side information: Exact recovery under the stochastic block model. *IEEE Journal of Selected Topics in Signal Processing*, 12(5):944–958, 2018.

Salakhutdinov, R. and Mnih, A. Probabilistic matrix factorization. In *Proceedings of the 20th International Conference on Neural Information Processing Systems*, pp. 1257–1264, 2007.

Salakhutdinov, R. and Mnih, A. Bayesian probabilistic matrix factorization using markov chain monte carlo. In *Proceedings of the 25th International Conference on Machine Learning*, pp. 880–887, 2008.

Sarwar, B., Karypis, G., Konstan, J., and Riedl, J. Item-based collaborative filtering recommendation algorithms. In *Proceedings of the 10th International Conference on World Wide Web*, pp. 285–295, 2001.

Traud, A. L., Mucha, P. J., and Porter, M. A. Social structure of facebook networks. *Physica A: Statistical Mechanics and its Applications*, 391(16):4165–4180, 2012.

Xu, J., Massoulié, L., and Lelarge, M. Edge label inference in generalized stochastic block models: from spectral theory to impossibility results. In *Proceedings of Machine Learning Research*, volume 35, pp. 903–920, 2014.

Yang, J., McAuley, J., and Leskovec, J. Community detection in networks with node attributes. In *2013 IEEE 13th International Conference on Data Mining*, pp. 1151–1156, 2013a.

Yang, X., Steck, H., Guo, Y., and Liu, Y. On top-k recommendation using social networks. In *Proceedings of the Sixth ACM Conference on Recommender Systems*, pp. 67–74, 2012.

Yang, X., Guo, Y., and Liu, Y. Bayesian-inference-based recommendation in online social networks. *IEEE Trans. Parallel Distrib. Syst.*, pp. 642–651, 2013b.

Yoon, J., Lee, K., and Suh, C. On the joint recovery of community structure and community features. In *56th Annual Allerton Conference on Communication, Control, and Computing*, pp. 686–694, 2018.

Yun, S.-Y. and Proutiere, A. Optimal cluster recovery in the labeled stochastic block model. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, pp. 973–981, 2016.

Zhang, Q., Suh, G., Suh, C., and Tan, V. Mc2g: An efficient algorithm for matrix completion with social and item similarity graphs. *arXiv preprint arXiv:2006.04373*, 2020.

Zhang, Q., Tan, V., and Suh, C. Community detection and matrix completion with social and item similarity graphs. *IEEE Transactions on Signal Processing*, 2021.