ONLINE LOW RANK MATRIX COMPLETION

Anonymous authors
Paper under double-blind review

ABSTRACT

We study the problem of *online* low-rank matrix completion with M users, N items and T rounds. In each round, the algorithm recommends one item per user, for which it gets a (noisy) reward sampled from a low-rank user-item preference matrix. The goal is to design a method with sub-linear regret (in T) and nearly optimal dependence on M and N. The problem can be easily mapped to the standard multi-armed bandit problem where each item is an *independent* arm, but that leads to poor regret as the correlation between arms and users is not exploited. On the other hand, exploiting the low-rank structure of reward matrix is challenging due to non-convexity of the low-rank manifold. We first demonstrate that the low-rank structure can be exploited using a simple explore-then-commit (ETC) approach that ensures a regret of $O(\mathsf{polylog}(\mathsf{M}+\mathsf{N})\mathsf{T}^{2/3})$. That is, roughly only $\mathsf{polylog}(\mathsf{M}+\mathsf{N})$ item recommendations are required per user to get a non-trivial solution. We then improve our result for the rank-1 setting which in itself is quite challenging and encapsulates some of the key issues. Here, we propose OCTAL (Online Collaborative filTering using iterAtive user cLustering) that guarantees nearly optimal regret of $O(\mathsf{polylog}(\mathsf{M}+\mathsf{N})\mathsf{T}^{1/2})$. OCTAL is based on a novel technique of clustering users that allows iterative elimination of items and leads to a nearly optimal minimax rate.

1 INTRODUCTION

Collaborative filtering based on low-rank matrix completion/factorization techniques are the cornerstone of most modern recommendation systems (Koren, 2008). Such systems model the underlying user-item affinity matrix as a low-rank matrix, use the acquired user-item recommendation data to estimate the low-rank matrix and subsequently, use the matrix estimate to recommend items for each user. Several existing works study this *offline* setting (Candès & Recht, 2009; Deshpande & Montanari, 2012; Jain et al., 2013; Chen et al., 2019; Abbe et al., 2020). However, typical recommendation systems are naturally *online* and interactive – they recommend items to users and need to adapt quickly based on users' feedback. The goal of such systems is to quickly identify each user's preferred set of items, so it is necessary to identify the best items for each user instead of estimating the entire affinity matrix. Moreover, items/users are routinely added to the system, so it should be able to quickly adapt to new items/users by using only a small amount of recommendation feedback.

In this work, we study this problem of the online recommendation system. In particular, we study the online version of low-rank matrix completion with the goal of identifying top few items for each user using say only logarithmic many exploratory recommendation rounds for each user. In each round (out of T rounds) we predict one item (out of N items) for each user (out of M users) and obtain feedback/reward for each of the predictions – e.g. did the users view the recommended movie. The goal is to design a method that has asymptotically similar reward to a method that can pick *best* items for each user. As mentioned earlier, we are specifically interested in the setting where $\mathsf{T} \ll \mathsf{N}$ i.e. the number of recommendation feedback rounds is much smaller than the total number of items.

Moreover, we assume that the expected reward matrix is low-rank. That is, if $\mathbf{R}_{ij}^{(t)}$ is the reward obtained in the tth round for predicting item-j for user-i, then $\mathbb{E}\mathbf{R}_{ij}^{(t)} = \mathbf{P}_{ij}$, where $\mathbf{P} \in \mathbb{R}^{\mathsf{M}\times\mathsf{N}}$ is a low-rank matrix. A similar low rank reward matrix setting has been studied in *online* multi-dimensional learning problems (Katariya et al., 2017b; Kveton et al., 2017; Trinh et al., 2020). But in these problems, the goal is to find the matrix entry with the highest reward. Instead, our goal is to recommend good items to *all* users which is a significantly harder challenge. A trivial approach is

to ignore the underlying low-rank structure and solve the problem using standard multi-arm bandit methods. That is, model each *(user, item)* pair as an *arm*. Naturally, that would imply exploration of almost all the items for *each* user, which is also reflected in the regret bound (averaged over users) of $O(\sqrt{\mathsf{NT}})$ (Remark 1). That is, as expected, the regret bound is vacuous when the number of recommendation rounds $\mathsf{T} \ll \mathsf{N}$.

In contrast, most of the existing online learning techniques that leverage structure amongst arms assume a parametric form for the reward function and require the reward function to be convex in the parameters (Shalev-Shwartz et al. 2011, Bubeck, 2011). Thus, due to the non-convexity of the manifold of low-rank matrices, such techniques do not apply in our case. While there are some exciting recent approaches for *non-convex online learning* (Agarwal et al. 2019; Suggala & Netrapalli, 2020), they do not apply to the above mentioned problem.

Our Techniques and Contributions: We first present a method based on the explore-then-commit (ETC) approach (Algorithm 2). For the first few rounds, the algorithm runs a pure exploration strategy by sampling random items for each user. We use the data obtained by pure exploration to learn a good estimate of the underlying reward matrix $\mathbf{P}$; this result requires a slight modification of the standard matrix completion result in Chen et al. (2019). We then run the exploitation rounds based on the current estimate. In particular, for the remaining rounds, the algorithm commits to the arm with the highest estimated reward for each user. With the ETC algorithm, we achieve a regret bound of $O(\mathsf{polylog}(\mathsf{M}+\mathsf{N})\mathsf{T}^{2/3})$ (Thm. 1). This bound is able to get rid of the dependence on N, implying non-trivial guarantees even when $\mathsf{T} \ll \mathsf{N}$. That is, we require only $\mathsf{polylog}(\mathsf{M}+\mathsf{N})$ exploratory recommendations per user. However, the dependence of the algorithm on T is sub-optimal. To address this, we study the special but critical case of rank-one reward matrix. The rank-1 setting is itself technically challenging (Katariya et al. 2017b) and encapsulates many key issues. We provide a novel algorithm OCTAL in Algorithm 3 and a modified version in Algorithm 8 that achieves nearly optimal regret bound of $O(\mathsf{polylog}(\mathsf{M}+\mathsf{N})\mathsf{T}^{1/2})$ (see Theorems 2 and E). The key insight is that in rank-one case, we need to cluster users based on their true latent representation to ensure low regret.

Our method OCTAL consists of multiple phases of exponentially increasing number of rounds. Each phase refines the current estimate of relevant sub-matrices of the reward matrix using standard matrix completion techniques. Using the latest estimate, we jointly refine the cluster of users and the estimate of the best items for users in each cluster. We can show that the regret in each phase decreases very quickly and since the count of phases required to get the correct clustering is small, we obtain our desired regret bounds. Finally, we show that our method achieves a regret guarantee that scales as $\widetilde{O}(\mathsf{T}^{1/2})$ with the number of rounds T. We also show that the dependence on T is optimal (see Theorem 3). Below we summarize our main contributions ($\widetilde{O}(\cdot)$ hides logarithmic factors):

- We formulate the online low rank matrix completion problem and define the appropriate notion of regret to study. We propose Algorithm 2 based on the explore-then-commit approach that suffers a regret of $\widetilde{O}(\mathsf{T}^{2/3})$. (Thm. 1)
- We propose a novel algorithm OCTAL (Online Collaborative filTering using iterAtive user cLustering) and a modified version (Alg. 3 and Alg. 8 respectively) for the special case of rank-1 reward matrices and guarantee a regret of $\widetilde{O}(\mathsf{T}^{1/2})$. Importantly, our algorithms provide non-trivial regret guarantees in the practical regime of $\mathsf{M} \gg \mathsf{T}$ and $\mathsf{N} \gg \mathsf{T}$ i.e when the number of users and items is much larger than the number of recommendation rounds. Moreover, OCTAL does not suffer an issue of large cold-start time (possibly large exploration period) as in the ETC algorithm.
- We conducted detailed empirical study of our proposed algorithms (see Appendix A) on synthetic and multiple real datasets, and demonstrate that our algorithms can achieve significantly lower regret than methods that do not use collaboration between users. Furthermore, we show that for rank-1 case, while it is critical to tune the exploration period in ETC (as a function of rounds and sub-optimality gaps) and is difficult in practice (Lattimore & Szepesvári, 2020)[Ch. 6], OCTAL still suffers lower regret without such side-information (see Figures 1a and 2b).

Technical Challenges: For rank-1 case, i.e., $\mathbf{P} = \mathbf{u}\mathbf{v}^\mathsf{T}$, one can cluster users in two bins: ones with $\mathbf{u}_i \geq 0$ and ones with $\mathbf{u}_i < 0$. Cluster-2 users dislike the best items for Cluster-1 users and vice-versa. Thus, we require algorithms that can learn this cluster structure and exploit the fact that within the same cluster, relative ranking of all items remain the same. This is a challenging information to exploit, and requires learning the latent structure. Note that an algorithm that first attempts to cluster all users and then optimize the regret will suffer the same $\mathsf{T}^{2/3}$ dependence in the worst case as in

the ETC algorithm; this is because the difficulty of clustering the users are widely different. Instead, our proposed OCTAL algorithm (Algorithms 3 and 8) in each iteration/phase performs two tasks: i) it tries to eliminate some of the items similar to standard phase elimination method (Lattimore & Szepesvári, 2020)[Ch 6, Ex 6.8] for users that are already clustered, ii) it simultaneously tries to grow the set of clustered users. For partial clustering, we first apply low rank matrix completion guarantees over carefully constructed reward sub-matrices each of which correspond to a cluster of users and a set of active items that have high rewards for all users in the same cluster, and then use the partial reward matrix for eliminating some of the items in each cluster.

1.1 Related Works

To the best of our knowledge, we provide first rigorous online matrix completion algorithms. But, there are several closely related results/techniques in the literature which we briefly survey below.

A very similar setting was considered in Sen et al. (2017) where the authors considered a multi-armed bandit problem with L contexts and K arms with context dependent reward distributions. The authors assumed that the $\mathsf{L} \times \mathsf{K}$ reward matrix is low rank and can be factorized into non-negative components which allowed them to use recovery guarantees from non-negative matrix factorization. Moreover, the authors only showed ETC algorithms that resulted in $\mathsf{T}^{2/3}$ regret guarantees. Our techniques can be used to improve upon the existing guarantees in Sen et al. (2017) in two ways 1) Removing the assumption of the low rank components being non-negative as we use matrix completion with entry-wise error guarantees. 2) The dependence on T can be improved from $\mathsf{T}^{2/3}$ to $\mathsf{T}^{1/2}$ when the reward matrix $\mathbf{P}$ is rank-1.

Multi-dimensional online decision making problems namely stochastic rank-1 matrix bandits was introduced in Katariya et al. (2017b,a); Trinh et al. (2020). In their settings, at each round $t \in [\mathsf{T}]$, the learning agent can choose one row and one column and observes a reward corresponding to an entry of a rank-1 matrix. Here, the regret is defined in terms of the best *(row ,column)* pair which corresponds to the best arm. This setting was extended to the rank r setting (Kveton et al., 2017), rank 1 multi-dimensional tensors (Hao et al., 2020), bilinear bandits (Jun et al., 2019; Huang et al., 2021) and generalized linear bandits (Lu et al., 2021). Although these papers provide tight regret guarantees, they cannot be translated to our problem. This is because, we solve a significantly different problem with an underlying rank-1 reward matrix $\mathbf{P}$ where we need to minimize the regret for all users (rows of $\mathbf{P}$) jointly. Hence, it is essential to find the entries (columns) of $\mathbf{P}$ with large rewards for each user(row) of $\mathbf{P}$; contrast this with the multi-dimensional online learning problem where it is sufficient to infer only the entry ((row,column) pair) in the matrix/tensor with the highest reward. Since the rewards for each user have different gaps, the analysis becomes involved for our OCTAL algorithm.

Another closely related line of work is the theoretical model for User-based Collaborative Filtering (CF) studied in Bresler et al. (2014; 2016); Heckel & Ramchandran (2017); Bresler & Karzand (2019); Huleihel et al. (2021). In particular, these papers were the first to motivate and theoretically analyze the collaborative framework with the restriction that the same item cannot be recommended more than once to the same user. Here, it is assumed that the M users can be partitioned into K latent clusters such that the preference vectors of users in the same cluster are close; the preference vectors of users in different clusters have small inner product. Such models are known to be restrictive and typically collaborative filtering via low-rank matrix completion based methods are preferred in real-world recommendation systems.

In the past decade, several papers have studied the problem of *offline* low rank matrix completion on its own (Mazumder et al., 2010; Negahban & Wainwright, 2012; Chen et al., 2019; Deshpande & Montanari, 2012; Abbe et al., 2020; Jain et al., 2013; Jain & Kar, 2017) and also in the presence of side information such as social graphs or similarity graphs (Xu et al., 2013; Ahn et al., 2018; 2021; Elmahdy et al., 2020; Jo & Lee, 2021; Zhang et al., 2022). Some of these results namely the ones that provide $\|\cdot\|_\infty$ norm guarantees on the estimated matrix can be adapted into Explore-Then-Commit (ETC) style algorithms (see Sec. 4). Finally, there is significant amount of related theoretical work for online non-convex learning (Suggala & Netrapalli, 2020; Yang et al., 2018; Huang et al., 2020) and empirical work for online Collaborative Filtering (Huang et al., 2020; Lu et al., 2013; Zhang et al., 2015) but they do not study the regret in online matrix completion setting.

Organization: In Section 2, we define the problem and the notion of regret that we minimize. In Section 3, we introduce several preliminary results (proofs delegated to Appendix B). In Section 4,

we discuss the Explore-Then-Commit (ETC) algorithm (proof in Appendix C) and in Section 5, we introduce and give an overview of the OCTAL algorithm (proofs delegated to Appendices D and E). In Appendix A, we provide detailed experimental results on synthetic and real world datasets.

2 Problem Definition

Notations: We write $[m]$ to denote the set $\{1, 2, \ldots, m\}$. For a vector $\mathbf{v} \in \mathbb{R}^m$, $\mathbf{v}_i$ denotes the ith element; for any set $\mathcal{U} \subseteq [m]$, let $\mathbf{v}_{\mathcal{U}}$ denote the vector $\mathbf{v}$ restricted to the indices in $\mathcal{U}$. $\mathbf{A}_i$ denotes the ith row of $\mathbf{A}$ and $\mathbf{A}_{ij}$ denotes the (i, j)-th element of matrix $\mathbf{A}$. $[n]$ denotes the set $\{1, 2, \ldots, n\}$. For any set $\mathcal{U} \subset [m], \mathcal{V} \subset [n]$, $\mathbf{A}_{\mathcal{U},\mathcal{V}}$ denotes the matrix $\mathbf{A}$ restricted to the rows in $\mathcal{U}$ and columns in $\mathcal{V}$. Also, let $\|\mathbf{A}\|_{2\to\infty}$ be the maximum ℓ_2 norm of the rows of $\mathbf{A}$ and $\|\mathbf{A}\|_\infty$ be the absolute value of the largest entry in $\mathbf{A}$. We write $\mathbb{E}X$ to denote the expectation of a random variable X.

Consider a system with a set of M users and N items. Let $\mathbf{P} = \mathbf{U}\mathbf{V}^\mathsf{T} \in \mathbb{R}^{\mathsf{M}\times\mathsf{N}}$ be the unknown reward matrix of rank $r < \min(\mathsf{M}, \mathsf{N})$ where $\mathbf{U} \in \mathbb{R}^{\mathsf{M}\times r}$ and $\mathbf{V} \in \mathbb{R}^{\mathsf{N}\times r}$ denote the latent embeddings corresponding to users and items respectively. In other words, we can denote $\mathbf{P}_{ij} \triangleq \langle \mathbf{u}_i, \mathbf{v}_j \rangle$ where $\mathbf{u}_i, \mathbf{v}_j \in \mathbb{R}^r$ denotes the r-dimensional embeddings of i-th user and the j-th item, respectively. Often, we will also use the SVD decomposition of $\mathbf{P} = \bar{\mathbf{U}}\Sigma\bar{\mathbf{V}}$ where $\bar{\mathbf{U}} \in \mathbb{R}^{\mathsf{M}\times r}, \bar{\mathbf{V}} \in \mathbb{R}^{\mathsf{N}\times r}$ are orthonormal matrices i.e. $\bar{\mathbf{U}}^\mathsf{T}\bar{\mathbf{U}} = \mathbf{I}$ and $\bar{\mathbf{V}}^\mathsf{T}\bar{\mathbf{V}} = \mathbf{I}$ and $\mathbf{\Sigma} \triangleq \mathsf{diag}(\lambda_1, \lambda_2, \ldots, \lambda_r) \in \mathbb{R}^{r\times r}$ is a diagonal matrix. We will denote the condition number of the matrix $\mathbf{P}$ by $\kappa \triangleq (\max_i \lambda_i)(\min_i \lambda_i)^{-1}$.

Consider a system that recommends one item to every user, in each round $t \in [\mathsf{T}]$. Let, $\mathbf{R}^{(t)}_{u\rho_u(t)}$ be the reward for recommending item $\rho_u(t) \in [\mathsf{N}]$ for user u. Also, let:

$$\mathbf{R}^{(t)}_{u\rho_u(t)} = \mathbf{P}_{u\rho_u(t)} + \mathbf{E}^{(t)}_{u\rho_u(t)} \tag{1}$$

where $\mathbf{E}^{(t)}_{u\rho_u(t)}$ denotes the unbiased additive noise. Each element of $\{\mathbf{E}^{(t)}_{u\rho_u(t)}\}_{u\in[\mathsf{M}],t\in[\mathsf{T}]}$ is assumed to be i.i.d. zero mean sub-gaussian random variables with variance proxy σ^2. That is, $\mathbb{E}[\mathbf{E}^{(t)}_{u\rho_u(t)}] = 0$ and $\mathbb{E}[\exp(s\mathbf{E}^{(t)}_{u\rho_u(t)})] \leq \exp(\sigma^2 s^2/2)$ for all $u \in [\mathsf{M}], t \in [\mathsf{T}]$. The goal is to minimize the expected regret where the expectation is over randomness in rewards and the algorithm:

$$\mathsf{Reg}(\mathsf{T}) \triangleq \frac{\mathsf{T}}{\mathsf{M}} \sum_{u\in[\mathsf{M}]} \max_{j\in[\mathsf{N}]} \mathbf{P}_{uj} - \mathbb{E}[\sum_{t\in[\mathsf{T}]} \frac{1}{\mathsf{M}} \sum_{u\in[\mathsf{M}]} \mathbf{R}^{(t)}_{u\rho_u(t)}]. \tag{2}$$

In this problem, the interesting regime is $(\mathsf{N}, \mathsf{M}) \gg \mathsf{T}$ as is often the case for most practical recommendation systems. Here, treating each user separately will lead to vacuous regret bounds as each item needs to be observed at least once by each user to find the best item for each user. However, low-rank structure of the rewards can help share information about items across users.

Remark 1. *If $\mathsf{T} \gg \mathsf{N}$, then we can treat each user as a separate multi-armed bandit problem. In that case, in our setting, the well-studied Upper Confidence Bound (UCB) algorithm achieves an expected regret of at most $O(\sigma\sqrt{\mathsf{N}\mathsf{T}\log\mathsf{T}})$ (Theorem 2.1 in Bubeck & Cesa-Bianchi (2012)).*

3 Preliminaries

Let us introduce a different observation model from (1). Consider an unknown rank r matrix $\mathbf{P} \in \mathbb{R}^{\mathsf{M}\times\mathsf{N}}$. For each entry $i \in [\mathsf{M}], j \in [\mathsf{N}]$, we observe:

$$\mathbf{P}_{ij} + \mathbf{E}_{ij} \text{ with probability } p, \quad 0 \text{ with probability } 1 - p, \tag{4}$$

where $\mathbf{E}_{ij}$ are independent zero mean sub-gaussian random variables with variance proxy $\sigma^2 > 0$. We now introduce the following result from Chen et al. (2019):

Lemma 1 (Theorem 1 in Chen et al. (2019))**.** *Let rank $r = O(1)$ matrix $\mathbf{P} \in \mathbb{R}^{d\times d}$ with SVD decomposition $\mathbf{P} = \bar{\mathbf{U}}\mathbf{\Sigma}\bar{\mathbf{V}}^\mathsf{T}$ satisfy $\|\bar{\mathbf{U}}\|_{2,\infty} \leq \sqrt{\mu r/d}, \|\bar{\mathbf{V}}\|_{2,\infty} \leq \sqrt{\mu r/d}$ and condition number $\kappa = O(1)$. Let $1 \geq p \geq C\mu^2 d^{-1}\log^3 d$ for some sufficiently large constant $C > 0$, $\sigma = O\left(\sqrt{\frac{pd}{\mu^3\log d}}\|\mathbf{P}\|_\infty\right)$. Suppose we observe noisy entries of $\mathbf{P}$ according to observation model*

Algorithm 1 ESTIMATE

Require: Set of users $\mathcal{U} \subseteq [\mathsf{M}]$, set of items $\mathcal{V} \subseteq [\mathsf{N}]$, total rounds m, set of indices $\Omega \subseteq \mathcal{U} \times \mathcal{V}$, rounds in each iteration $b = \max_{u \in \mathcal{U}} |v \in \mathcal{V} \mid (u, v) \in \Omega|$, regularization parameter λ. Index of round t is relative to the first round when the algorithm is invoked; hence $t = 1, 2, \ldots, m$.

1: **for** $\ell = 1, 2, \ldots, m/b$ **do**
2: For all $(i, j) \in \Omega$, set $\mathsf{Mask}_{ij} = 0$.
3: **for** $\ell' = 1, 2, \ldots, b$ **do**
4: **for** each user $u \in \mathcal{U}$ in round $t = (\ell - 1)b + \ell'$ **do**
5: Recommend an item $\rho_u(t)$ in $\{j \in \mathcal{V} \mid (u, j) \in \Omega, \mathsf{Mask}_{uj} = 0\}$ and set $\mathsf{Mask}_{u\rho_u(t)} = 1$. If not possible then recommend any item $\rho_u(t)$ in $\mathcal{V}$ s.t. $(u, \rho_u(t)) \notin \Omega$. Observe $\mathbf{R}^{(t)}_{u\rho_u(t)}$.
6: **end for**
7: **end for**
8: **end for**
9: For each $(u, j) \in \Omega$, compute $\mathbf{Z}_{uj}$ to be average of $\lfloor m/b \rfloor$ observations corresponding to user u being recommended item j i.e. $\mathbf{Z}_{uj} = \text{avg}\{\mathbf{R}^{(t)}_{u\rho_u(t)}$ for $t \in [m] \mid \rho_u(t) = j\}$. Discard all other observations corresponding to indices not in Ω.
10: Without loss of generality, assume $|\mathcal{U}| \leq |\mathcal{V}|$. For each $i \in \mathcal{V}$, independently set δ_i to be a value in the set $[\lceil |\mathcal{V}|/|\mathcal{U}| \rceil]$ uniformly at random. Partition indices in $\mathcal{V}$ into $\mathcal{V}^{(1)}, \mathcal{V}^{(2)}, \ldots, \mathcal{V}^{(k)}$ where $k = \lceil |\mathcal{V}|/|\mathcal{U}| \rceil$ and $\mathcal{V}^{(q)} = \{i \in \mathcal{V} \mid \delta_i = q\}$ for each $q \in [k]$. Set $\Omega^{(q)} \leftarrow \Omega \cap (\mathcal{U} \times \mathcal{V}^{(q)})$ for all $q \in [k]$. *#If $|\mathcal{U}| \geq |\mathcal{V}|$, we partition the indices in $\mathcal{U}$.*
11: **for** $q \in [k]$ **do**
12: Solve convex program

$$\min_{\mathbf{Q}^{(q)} \in \mathbb{R}^{|\mathcal{U}| \times |\mathcal{V}^{(q)}|}} \frac{1}{2} \sum_{(i,j) \in \Omega^{(q)}} (\mathbf{Q}^{(q)}_{i\pi(j)} - \mathbf{Z}_{ij})^2 + \lambda \|\mathbf{Q}^{(q)}\|_\star, \tag{3}$$

where $\|\mathbf{Q}^{(q)}\|_\star$ denotes nuclear norm of matrix $\mathbf{Q}^{(q)}$ and $\pi(j)$ is index of j in set $\mathcal{V}^{(q)}$.
13: **end for**
14: Return $\widetilde{\mathbf{Q}} \in \mathbb{R}^{\mathsf{M} \times \mathsf{N}}$ s.t. $\widetilde{\mathbf{Q}}_{\mathcal{U}, \mathcal{V}^{(q)}} = \mathbf{Q}^{(q)}$ for all $q \in [k]$ and for every $(i, j) \notin \mathcal{U} \times \mathcal{V}$, $\widetilde{\mathbf{Q}}_{ij} = 0$.

in (4). Then, with probability exceeding $1 - O(d^{-3})$, we can compute a matrix $\widehat{\mathbf{P}}$ by using Algorithm 4 (Appendix B) with parameters ($\mathcal{U} = [\mathsf{M}], \mathcal{V} = [\mathsf{N}], \sigma^2, r, p$) s.t.,

$$\|\widehat{\mathbf{P}} - \mathbf{P}\|_\infty \leq O\Big(\frac{\sigma}{\min_i \lambda_i} \cdot \sqrt{\frac{\mu d \log d}{p}} \|\mathbf{P}\|_\infty\Big). \tag{5}$$

Note there are several difficulties in using Lemma 1 directly in our setting which are discussed below:

Remark 2 (Matrix Completion for Rectangular Matrices)**.** *Lemma 1 is described for square matrices and a trivial approach to use Lemma 1 for rectangular matrices with M rows and N columns (say $\mathsf{N} \geq \mathsf{M}$) by appending $\mathsf{N} - \mathsf{M}$ zero rows leads to an undesirable $(\mathsf{N}/\mathsf{M})^{1/2}$ factor (Lemma 5) in the error bound (the $(\mathsf{N}/\mathsf{M})^{1/2}$ factor does not arise if we care about spectral/Frobenius norm instead of L_∞ norm). One way to resolve the issue is to partition the columns into N/M groups by assigning each column into one of the groups uniformly at random. Thus, we create N/M matrices which are almost square and apply Lemma 1 to recover an estimate that is close in L_∞ norm. Thus we can recover an estimate of the entire matrix which is close in L_∞ norm up to the desired accuracy without suffering the undesirable $(\mathsf{N}/\mathsf{M})^{1/2}$ factor (Lemma 6 and Steps 10-12 in Algorithm 1).*

Remark 3 (Observation Models)**.** *The observation model in equation 4 is significantly different from equation 1. In the former, a noisy version of each element of $\mathbf{P}$ is observed independently with probability p while in the latter, in each round $t \in [\mathsf{T}]$, for each user $u \in [\mathsf{M}]$, we observe noisy version of a chosen element $\rho_u(t)$. Our approach to resolve this discrepancy theoretically is to first sample a set Ω of indices according to equation 4 and subsequently use equation 1 to observe the indices in Ω (see Steps 3-6 in Algorithm 1 and Corollary 1). Of course, this implies obtaining observations corresponding to indices in a super-set of Ω (see Step 5 in Algorithm 1) and only using the observations in Ω for obtaining an estimate of the underlying matrix. In practice, this is not necessary and we can use all the observed indices to obtain the estimate in Step 12 of Algorithm 1.*

Remark 4 (Repetition and Median Tricks)**.** *The smallest error that is possible to achieve by using Lemma 1 is by substituting $p = 1$ and thereby obtaining $\|\widehat{\mathbf{P}} - \mathbf{P}\|_\infty \leq O\Big(\sigma(\min_i \lambda_i)^{-1} \cdot$*

$\sqrt{\mu d \log d}\|\mathbf{P}\|_\infty\Big)$ *and moreover, the probability of failure is polynomially small in the dimension* d*; however, this is insufficient when* d *is not large enough. Two simple tricks allow us to resolve this issue: 1) First we can obtain repeated observations from the same entry of the reward matrix and take its average;* s *repetitions can bring down the noise variance to* σ^2/s *2) Second, we can use the median trick where we obtain several independent estimates of the reward matrix and compute the element-wise median to boost the success probability (see proof of Lemma 2).*

We address all these issues (see Appendix B for detailed proofs) and arrive at the following lemma:

Lemma 2. *Let rank* $r = O(1)$ *reward matrix* $\mathbf{P} \in \mathbb{R}^{\mathsf{M}\times\mathsf{N}}$ *with SVD decomposition* $\mathbf{P} = \bar{\mathbf{U}}\bar{\mathbf{\Sigma}}\bar{\mathbf{V}}^\mathsf{T}$ *satisfy* $\|\bar{\mathbf{U}}\|_{2,\infty} \leq \sqrt{\mu r/\mathsf{M}}, \|\bar{\mathbf{V}}\|_{2,\infty} \leq \sqrt{\mu r/\mathsf{N}}$ *and condition number* $\kappa = O(1)$*. Let* $d_1 = \max(\mathsf{M},\mathsf{N})$*,* $d_2 = \min(\mathsf{M},\mathsf{N})$ *such that* $d_2 = \Omega(\mu r \log(r d_2))$ *and* $1 \geq p \geq C\mu^2 d_2^{-1}\log^3 d_2$ *for sufficiently large constant* $C > 0$*. Suppose we observe noisy entries of* $\mathbf{P}$ *according to observation model in (1). For any positive integer* $s > 0$ *satisfying* $\frac{\sigma}{\sqrt{s}} = O\Big(\sqrt{\frac{pd_2}{\mu^3\log d_2}}\|\mathbf{P}\|_\infty\Big)$*, there exists an algorithm* $\mathcal{A}$ *with parameters* s, p, σ *that uses* $m = O\Big(s\log(\mathsf{MN}\delta^{-1})(\mathsf{N}p + \sqrt{\mathsf{N}p\log\mathsf{M}\delta^{-1}})\Big)$ *rounds to compute a matrix* $\widehat{\mathbf{P}}$ *such that with probability exceeding* $1 - O(\delta\log(\mathsf{MN}\delta^{-1}))$

$$\|\mathbf{P} - \widehat{\mathbf{P}}\|_\infty \leq O\Big(\frac{\sigma r}{\sqrt{s d_2}}\sqrt{\frac{\mu^3\log d_2}{p}}\Big). \tag{6}$$

Remark 5. *Alg.* $\mathcal{A}$ *repeats the following process* $O(\log(\mathsf{MN}\delta^{-1}))$ *times: 1) sample subset of indices* $\Omega \subseteq [\mathsf{M}]\times[\mathsf{N}]$ *such that every* $(i,j) \in [\mathsf{M}]\times[\mathsf{N}]$ *is inserted into* Ω *independently with probability* p*. 2) By setting* $b = \max_{i\in[\mathsf{M}]}|j \in [\mathsf{N}] \mid (i,j) \in \Omega|$*, Algorithm* $\mathcal{A}$ *invokes Alg. 1 with total rounds* bs*, number of rounds in each iteration* b*, set* Ω*, set of users* $[\mathsf{M}]$*, items* $[\mathsf{N}]$ *and regularization parameter* $\lambda = C_\lambda\sigma\sqrt{\min(\mathsf{M},\mathsf{N})p}$ *for a suitable constant* $C_\lambda > 0$ *in order to compute an estimate of* $\mathbf{P}$*. The final estimate* $\widehat{\mathbf{P}}$ *is computed by taking an entry-wise median of each individual estimates obtained as output from each invocation of Alg. 1. Alternatively, Alg.* $\mathcal{A}$ *is detailed in Alg. 7 in Appendix B.*

Note that the total number of noisy observations made from the matrix $\mathbf{P}$ is $m \cdot \mathsf{M} \geq \mathsf{MN}\cdot p\cdot s$. Therefore, informally speaking, the average number of observations per index is $p \cdot s$ which results in an error of $\widetilde{O}(\sigma/\sqrt{sp})$ ignoring other terms (contrast with error $\widetilde{O}(\sigma/\sqrt{p})$ in equation 5.)

Remark 6 (Setting parameters s, p in lemma 2)**.** *Lemma 2 has three input parameters namely* $s \in \mathbb{Z}, 0 \leq p \leq 1$ *and* $0 \leq \delta \leq 1$*. For any set of input parameters* (η, ν)*, our goal is to set* s, p, δ *as functions of known* σ, r, μ, d_2 *such that we can recover* $\|\mathbf{P} - \widehat{\mathbf{P}}\|_\infty \leq \eta$ *with probability* $1-\nu$ *for which the conditions on* σ *and* p *are satisfied. From (17), we must have* $\sqrt{sp} = \frac{c\sigma r}{\sqrt{d_2}}\frac{\sqrt{\mu^3\log d_2}}{\eta}$ *for some appropriate constant* $c > 0$*. If* $r = O(1)$ *and* $\eta \leq \|\mathbf{P}\|_\infty$*, then an appropriate choice of* c *also satisfies the condition* $\frac{\sigma}{\sqrt{s}} = O\Big(\sqrt{\frac{pd_2}{\mu^3\log d_2}}\|\mathbf{P}\|_\infty\Big)$*. More precisely, we are going to set* $p = C\mu^2 d_2^{-1}\log^3 d_2$ *and* $s = \left\lceil\left(\frac{c\sigma r\sqrt{\mu}}{\eta\log d_2}\right)^2\right\rceil$ *in order to obtain the desired guarantee.*

4 Explore-Then-Commit (ETC) Algorithm

In this section, we present an Explore-Then-Commit (ETC) based algorithm for online low-rank matrix completion. The algorithm has two disjoint phases of exploration and exploitation. We will first jointly explore the set of items for all users for a certain number of rounds and compute an estimate $\widehat{\mathbf{P}}$ of the reward matrix $\mathbf{P}$. Subsequently, we commit to the estimated best item found for each user and sample the reward of the best item for the remaining rounds in the exploitation phase for that user. Note that the exploration phase involves using a matrix completion estimator in order to estimate the entire reward matrix $\mathbf{P}$ from few observed entries. Our regret guarantees in this framework is derived by carefully balancing exploration phase length and the matrix estimation error.

Theorem 1. *Consider the rank-*r *online matrix completion problem with* M *users,* N *items,* T *recommendation rounds. Set* $d_2 = \min(\mathsf{M},\mathsf{N})$*. Let* $\mathbf{R}^{(t)}_{u\rho_u(t)}$ *be the reward in each round, defined as in equation 1. Suppose* $d_2 = \Omega(\mu r\log(rd_2))$*. Let* $\mathbf{P} \in \mathbb{R}^{\mathsf{M}\times\mathsf{N}}$ *be the expected reward matrix*

Algorithm 2 ETC ALGORITHM

Require: users M, items N, rounds T, noise σ^2, rank r of $\mathbf{P}$, number of estimates $f = O(\log \mathsf{T})$.

Set $d_2 = \min(\mathsf{M}, \mathsf{N})$ and $v = (\mathsf{N}\|\mathbf{P}\|_\infty)^{-2/3}\left(\frac{\mathsf{T}\sigma r}{\sqrt{d_2}}\sqrt{\mu^3 \log d_2}\right)^{2/3}$. Set $p = C\mu^2 d_2^{-1}\log^3 d_2$, $s = \lceil vp^{-1}\rceil$ and $\lambda = C_\lambda \sigma\sqrt{d_2 p}$ for some constants $C, C_\lambda > 0$.
for $k = 1, 2, \ldots, f$ **do**
  For each tuple of indices $(i, j) \in [\mathsf{M}] \times [\mathsf{N}]$, independently set $\delta_{ij} = 1$ with probability p and $\delta_{ij} = 0$ with probability $1 - p$.
  Denote $\Omega = \{(i, j) \in [\mathsf{M}] \times [\mathsf{N}] \mid \delta_{ij} = 1\}$ and $b = \max_{i\in[\mathsf{M}]} \mid |j \in [\mathsf{N}] \mid (i, j) \in \Omega|$ to be the maximum number of index tuples in a particular row. Set total number of rounds to be bs.
  Compute the kth estimate $\widehat{\mathbf{P}}^{(k)} = \text{ESTIMATE}([\mathsf{M}], [\mathsf{N}], bs, \Omega, b, \lambda)$. # *(Algorithm 1 is used to recommend items to every user for bs rounds.*
end for
Compute final estimate estimate $\widehat{\mathbf{P}}$ by taking the entry-wise median of $\widehat{\mathbf{P}}^{(1)}, \widehat{\mathbf{P}}^{(2)}, \ldots, \widehat{\mathbf{P}}^{(f)}$.
for each of remaining rounds **do**
  Recommend $\mathsf{argmax}_{j\in[\mathsf{N}]}\widehat{\mathbf{P}}_{ij}$ for each user $i \in [\mathsf{M}]$. # *Number of remaining rounds is* $\mathsf{T} - bsf$.
end for

that satisfies the conditions stated in Lemma 2, and let σ^2 be the noise variance in rewards. Then, Algorithm 2 applied to the online rank-r matrix completion problem guarantees the following regret:

$$\mathsf{Reg}(\mathsf{T}) = O\Big(\Big(\mathsf{T}^{\frac{2}{3}}(\sigma^2 r^2\|\mathbf{P}\|_\infty)^{\frac{1}{3}}\Big(\frac{\mu^3\mathsf{N}\log d_2}{d_2}\Big)^{1/3} + \frac{\mathsf{N}\mu^2}{d_2}\log^3 d_2\Big)\log^2(\mathsf{MNT}) + \frac{\|\mathbf{P}\|_\infty}{\mathsf{T}^2}\Big). \quad (7)$$

Remark 7 (Non-trivial regret bounds)**.** *Theorem 1 provides non-trivial regret guarantees in the key regime when $\mathsf{N} \gg \mathsf{T}$ and $\mathsf{M} > \mathsf{N}$ where the regret scales only logarithmically on M, N. This is intuitively satisfying since in each round we are obtaining M observations, so more users translate to more information which in-turn allows better understanding of the underlying reward matrix. However, the dependence of regret on T (namely $\mathsf{T}^{2/3}$) is sub-optimal. In the subsequent section, we provide a novel algorithm to obtain regret guarantees with $\mathsf{T}^{1/2}$ for rank-1 $\mathbf{P}$.*

Remark 8 (Gap dependent bounds)**.** *Define the minimum gap to be $\Delta = \min_{u\in[\mathsf{M}]}\left|\mathbf{P}_{u\pi_u(1)} - \mathbf{P}_{u\pi_u(2)}\right|$ where $\pi_u(1), \pi_u(2)$ corresponds to the items with the highest and second highest reward for user u respectively. If the quantity Δ is known then it is possible to design ETC algorithms where the exploration phase is tuned accordingly in order to obtain regret bounds that scale logarithmically with the number of rounds T.*

5 OCTAL ALGORITHM

In this section we present our algorithm OCTAL (Algorithm 3) for online matrix completion where the reward matrix $\mathbf{P}$ is rank 1. The set of users is described by a latent vector $\mathbf{u} \in \mathbb{R}^{\mathsf{M}}$ and the set of items is described by a latent vector $\mathbf{v} \in \mathbb{R}^{\mathsf{N}}$. Thus $\mathbf{P} = \mathbf{u}\mathbf{v}^\mathsf{T}$ with SVD decomposition $\mathbf{P} = \bar{\lambda}\bar{\mathbf{u}}\bar{\mathbf{v}}^\mathsf{T}$.

Algorithm Overview: Our first key observation is that as $\mathbf{P}$ is rank-one, we can partition the set of users into two disjoint clusters $\mathcal{C}_1, \mathcal{C}_2$ where $\mathcal{C}_1 \equiv \{i \in [\mathsf{M}] \mid \mathbf{u}_i \geq 0\}$ and $\mathcal{C}_2 \equiv [\mathsf{M}] \setminus \mathcal{C}_1$. Clearly, for all users $u \in \mathcal{C}_1$, the item that results in maximum reward is $j_{\max} = \mathsf{argmax}_{t\in[\mathsf{N}]}\mathbf{v}_t$. On the other hand, for all users $u \in \mathcal{C}_2$, the item that results in maximum reward is $j_{\min} = \mathsf{argmin}_{t\in[\mathsf{N}]}\mathbf{v}_t$. Thus, if we can identify $\mathcal{C}_1, \mathcal{C}_2$ and estimate items with high reward (identical for users in the same cluster) using few recommendations per user, we can ensure low regret.

But, initially $\mathcal{C}_1, \mathcal{C}_2$ are unknown, so all users are *unlabelled* i.e., their cluster is unknown. In each phase (the outer loop indexed by ℓ), Algorithm 3 tries to label at least a few unlabelled users correctly. This is achieved by progressively refining estimate $\widetilde{\mathbf{Q}}$ of the reward matrix $\mathbf{P}$ restricted to the unlabelled users and all items (Step 12). Subsequently, unlabelled users for which the difference in maximum and minimum reward (inferred from estimated reward matrix) is large are labelled (Step 19). At the same time, in Step 13 users labelled in previous phases are partitioned into two clusters (denoted by $\mathcal{M}^{(\ell,1)}$ and $\mathcal{M}^{(\ell,2)}$) and for each of them, the algorithm refines an estimate of two distinct sub-matrices of the reward matrix $\mathbf{P}$ by recommending items only from a refined set ($\mathcal{N}^{(\ell,1)}$ and $\mathcal{N}^{(\ell,2)}$ respectively) containing the best item ($j_{\max}$ or $j_{\min}$). We also identify a small set

Algorithm 3 OCTAL (ONLINE COLLABORATIVE FILTERING USING ITERATIVE USER CLUSTERING)

Require: Number of users M, items N, rounds T, noise σ^2

1: Set $\mathcal{M}^{(1,1)} = \mathcal{M}^{(1,2)} = \phi$ and $\mathcal{B}^{(1)} = [\mathsf{M}]$. Set $\mathcal{N}^{(1,1)} = \mathcal{N}^{(1,2)} = \phi$. Set $f = O(\log(\mathsf{MNT}))$ and suitable constants $a, c, C, C', C_\lambda > 0$.
2: **for** $\ell = 1, 2, \ldots,$ **do**
3: Set $\Delta_\ell = C' 2^{-\ell} \min\left(\|\mathbf{P}\|_\infty, \frac{\sigma\sqrt{\mu}}{\log \mathsf{N}}\right)$.
4: **for** $k = 1, 2, \ldots, f$ **do**
5: **for** each pair of non-null sets $(\mathcal{B}^{(\ell)}, \mathsf{N})$, $(\mathcal{M}^{(\ell,1)}, \mathcal{N}^{(\ell,1)})$, $(\mathcal{M}^{(\ell,2)}, \mathcal{N}^{(\ell,2)}) \subseteq [\mathsf{M}] \times [\mathsf{N}]$ **do**
6: Denote $(\mathcal{T}^{(1)}, \mathcal{T}^{(2)})$ to be the considered pair of sets and $i \in \{0, 1, 2\}$ to be its index.
7: Set $d_{2,i} = \min(|\mathcal{T}^{(1)}|, |\mathcal{T}^{(2)}|)$. Set $p_{\ell,i} = C\mu^2 d_{2,i}^{-1} \log^3 d_{2,i}$ and $s_{\ell,i} = \left\lceil \left(\frac{c\sigma\sqrt{\mu}}{\Delta_\ell \log d_{2,i}}\right)^2 \right\rceil$.
8: For each tuple of indices $(u, v) \in \mathcal{T}^{(1)} \times \mathcal{T}^{(2)}$, independently set $\delta_{uv} = 1$ with probability $p_{\ell,i}$ and $\delta_{uv} = 0$ with probability $1 - p_{\ell,i}$.
9: Denote $\Omega^{(i)} = \{(u, v) \in \mathcal{T}^{(1)} \times \mathcal{T}^{(2)} \mid \delta_{uv} = 1\}$ and $b_{\ell,i} = \max_{u \in \mathcal{U}} |v \in \mathcal{V} \mid (u, v) \in \Omega|$. Set total number of rounds to be $m_{\ell,i} = b_{\ell,i} s_{\ell,i}$.
10: **end for**
11: Set $m_\ell = \max_{i \in \{0,1,2\}} m_{\ell,i}$.
12: Compute $\widetilde{\mathbf{Q}}^{(\ell,k)} = \text{ESTIMATE}(|\mathcal{B}^{(\ell)}|, [\mathsf{N}], m_\ell, \Omega^{(0)}, b_{\ell,0}, \lambda = C_\lambda \sigma \sqrt{d_{2,0} p_\ell})$. *# Algorithm 1 is used to recommend items to every user in $\mathcal{B}^{(\ell)}$ for m_ℓ rounds.*
13: For $i \in \{1, 2\}$, compute $\widetilde{\mathbf{P}}^{(\ell,i,f)} = \text{ESTIMATE}(|\mathcal{M}^{(\ell,i)}|, |\mathcal{N}^{(\ell,i)}|, m_\ell, \Omega^{(i)}, b_{\ell,i}, \lambda = C_\lambda \sigma \sqrt{d_{2,i} p_\ell})$. *# Algorithm 1 recommends items to every user in $\mathcal{M}^{(\ell,i)}$ for m_ℓ rounds.*
14: **end for**
15: Compute $\widetilde{\mathbf{Q}}^{(\ell)}$ =Entrywise Median$(\{\widetilde{\mathbf{Q}}^{(\ell,k)}\}_{k=1}^f)$, $\widetilde{\mathbf{P}}^{(\ell,i)}$ =Entrywise Median$(\{\widetilde{\mathbf{P}}^{(\ell,i,k)}\}_{k=1}^f)$ for $i \in \{1, 2\}$.
16: Set $\mathcal{B}^{(\ell+1)} \equiv \left\{u \in \mathcal{B}^{(\ell)} \mid \left|\max_{t \in [\mathsf{N}]} \widetilde{\mathbf{Q}}_{ut}^{(\ell)} - \min_{t \in [\mathsf{N}]} \widetilde{\mathbf{Q}}_{ut}^{(\ell)}\right| \leq 2a\Delta_\ell\right\}$
17: Compute $\mathcal{T}_u^{(\ell+1)} = \{j \in [\mathsf{N}]\} \mid \widetilde{\mathbf{Q}}_{uj}^{(\ell)} + \Delta_\ell > \max_{t \in [\mathsf{N}]} \widetilde{\mathbf{Q}}_{ut}^{(\ell)}\}$ for all $u \in \mathcal{B}^{(\ell)} \setminus \mathcal{B}^{(\ell+1)}$.
18: For $i \in \{1, 2\}$, for all users $u \in \mathcal{M}^{(\ell,i)}$, compute $\mathcal{T}_u^{(\ell+1)} = \{j \in \mathcal{N}^{(\ell,i)} \mid \widetilde{\mathbf{P}}_{uj}^{(\ell,i)} + \Delta_\ell > \max_{t \in \mathcal{N}^{(\ell,i)}} \widetilde{\mathbf{P}}_{ut}^{(\ell,i)}\}$.
19: Set v to be any user in $[\mathsf{M}] \setminus \mathcal{B}^{(\ell+1)}$. Set $\mathcal{M}^{(\ell+1,1)} = \{u \in [\mathsf{M}] \setminus \mathcal{B}^{(\ell+1)} \mid \mathcal{T}_u^{(\ell+1)} \cap \mathcal{T}_v^{(\ell+1)} \neq \phi\}$. Set $\mathcal{M}^{(\ell+1,2)} = [\mathsf{M}] \setminus (\mathcal{B}^{(\ell+1)} \cup \mathcal{M}^{(\ell+1,1)})$.
20: Compute $\mathcal{N}^{(\ell+1,1)} = \bigcap_{u \in \mathcal{M}^{(\ell+1,1)}} \mathcal{T}_u^{(\ell+1)}$, $\mathcal{N}^{(\ell+1,2)} = \bigcap_{u \in \mathcal{M}^{(\ell+1,2)}} \mathcal{T}_u^{(\ell+1)}$.
21: For $i \in \{1, 2\}$, if $|\mathcal{M}^{(\ell+1,i)}| \leq \frac{\mathsf{M}}{\sqrt{\mathsf{T}}}$, then set $\mathcal{B}^{(\ell+1)} \leftarrow \mathcal{B}^{(\ell+1)} \cup \mathcal{M}^{(\ell+1,i)}$ and $\mathcal{M}^{(\ell+1,i)} \leftarrow \phi$.
22: **end for**

of *good* items for each labelled user (including users labelled in previous phases), which correspond to large estimated rewards. We partition all these users into two clusters ($\mathcal{M}^{(\ell+1,1)}$ and $\mathcal{M}^{(\ell+1,2)}$) such that the set of *good* items for users in different clusters are disjoint. We can prove that such a partitioning is possible; users in same cluster have same sign of user embedding.

We also prove that the set of good items contain the best item for each labelled user ($j_{\max}$ or $j_{\min}$). So, after each phase, for each cluster of users, we compute the intersection of *good* items over all users in the cluster. This subset of items (*joint good* items) must contain the best item for that cluster and therefore we can discard the other items (Step 20). We can show that all items in the set of *joint good* items ($\mathcal{N}^{(\ell+1,1)}$ and $\mathcal{N}^{(\ell+1,2)}$) have rewards which is close to the reward of the best item. Therefore the algorithm suffers small regret if for each group of labelled users, the algorithm recommends items from the set of *joint good* items (Step 13) in the next phase. We can further show that for the set of unlabelled users, the difference in rewards between the best item and worst item is small and hence the regret for such users is small, irrespective of the recommended item (Step 12). Note that until the number of labelled users is sufficiently large, we do not consider them separately (Step 21). A crucial part of our analysis is to show that for any subset of users and items considered in Step 5, the number of rounds sufficient to recover a good estimate of the expected reward sub-matrix is small irrespective of the number of considered items (if the number of users is sufficiently large).

Remark 9 (Practical considerations)**.** *In general OCTAL (Alg. 3) is computationally faster than the ETC Algorithm (Alg. 2) with a higher exploration length. This is because OCTAL eliminates large*

chunks of items in every phase and therefore has to solve easier optimization problems; on the other hand, ETC has to solve a low rank matrix completion problem in MN *variables that becomes slower with the exploration length (datapoints). Moreover, OCTAL algorithm runs in phases with the initial phases being very small; hence the users do not have to wait for a long time to even get personalized recommendations like in ETC. These features make OCTAL much more practical than ETC.*

To summarize, in Algorithm 3, the entire set of rounds $[\mathsf{T}]$ is partitioned into phases of exponentially increasing length. In each phase, for the set of unlabelled users, we do pure exploration and recommend random items from the set of all possible items (Step 12). The set of labelled users are partitioned into two clusters; for each, we follow a semi-exploration strategy where we recommend random items from a set of *joint good* items (Steps 13). We now introduce the following definition:

Definition 3 ((α, μ)-Local Incoherence)**.** *For* $0 \leq \alpha \leq 1$*, a vector* $\mathbf{v} \in \mathbb{R}^m$ *is* (α, μ)*-local incoherent if for all sets* $\mathcal{U} \subseteq [m]$ *satisfying* $|\mathcal{U}| \geq \alpha m$*, we have* $\|\mathbf{v}_{\mathcal{U}}\|_\infty \leq \sqrt{\frac{\mu}{|\mathcal{U}|}}\|\mathbf{v}_{\mathcal{U}}\|_2$.

Note that the local incoherence condition is trivially satisfied if the magnitude of each vector entry is bounded from below. We are now ready to state our main result:

Theorem 2. *Consider the rank-*1 *online matrix completion problem with* T *rounds,* M *users s.t.* $\mathsf{M} \geq \sqrt{\mathsf{T}}$ *and* N *items. Denote* $d_2 = \min(\mathsf{M}, \mathsf{N})$*. Let* $\mathbf{R}^{(t)}_{u\rho_u(t)}$ *be the reward in each round, defined as in equation 1. Let* σ^2 *be the noise variance in rewards and let* $\mathbf{P} \in \mathbb{R}^{\mathsf{M}\times\mathsf{N}}$ *be the expected reward matrix with SVD decomposition* $\mathbf{P} = \lambda\bar{\mathbf{u}}\bar{\mathbf{v}}^\mathsf{T}$ *such that* $\bar{\mathbf{u}}$ *is* $(\mathsf{T}^{-1/2}, \mu)$*-locally incoherent,* $\|\bar{\mathbf{v}}\|_\infty \leq \sqrt{\mu/\mathsf{N}}$*,* $d_2 = \Omega(\mu \log d_2)$ *and* $|\bar{\mathbf{v}}_{j_{\min}}| = \Theta(|\bar{\mathbf{v}}_{j_{\max}}|)$*. Then, by suitably choosing parameters* $\{\Delta_\ell\}_\ell$*, positive integers* $\{s_{(\ell,0)}, s_{(\ell,1)}, s_{(\ell,2)}\}_\ell$ *and* $1 \geq \{p_{(\ell,0)}, p_{(\ell,1)}, p_{(\ell,2)}\}_\ell \geq 0$ *as described in Algorithm 3, we can ensure a regret guarantee of* $\mathsf{Reg}(\mathsf{T}) = O(\sqrt{\mathsf{T}}\|\mathbf{P}\|_\infty + \mathsf{J}\sqrt{\mathsf{TV}})$ *where* $\mathsf{J} = O\Big(\log\Big(\frac{1}{\sqrt{\mathsf{VT}^{-1}}}\min\Big(\|\mathbf{P}\|_\infty, \frac{\sigma\sqrt{\mu}}{\log \mathsf{N}}\Big)\Big)\Big)$ *and* $\mathsf{V} = \Big(\max(1, \frac{\mathsf{N}\sqrt{\mathsf{T}}}{\mathsf{M}})\sigma^2\mu^3\log^2(\mathsf{MNT})\Big)$.

Similar to Algorithm 2, Algorithm 3 allows non-trivial regret guarantees even when $\mathsf{N} \gg \mathsf{T}$ provided the number of users is significantly large as well i.e. $\mathsf{M} = \widetilde{\Omega}(\mathsf{N}\sqrt{\mathsf{T}})$.

Remark 10. *Under slightly stronger local incoherence conditions on the vector* $\bar{\mathbf{u}}$*, we can analyze a modified version of* OCTAL *(Alg. 8) without requiring users* M *to be large. When* $|\mathcal{C}_1| \approx |\mathcal{C}|_2$*, the regret guarantee (Thm. E) scales as* $\widetilde{O}(\sqrt{\mathsf{NT}/\mathsf{M}})$*. Due to space limitations, details of Algorithm 8 and the proof of Theorem E can be found in Appendix E.*

Finally, we show that the above dependence on $\mathsf{N}, \mathsf{M}, \mathsf{T}$ matches the lower bound that we obtain by reduction to the well-known multi-armed bandit problem.

Theorem 3. *Let* $\mathbf{P} \in [0, 1]^{\mathsf{M}\times\mathsf{N}}$ *be a rank* 1 *reward matrix and the noise variance* $\sigma^2 = 1$*. In that case, any algorithm for online matrix completion problem will suffer regret of* $\Omega(\sqrt{\mathsf{NTM}^{-1}})$.

6 Conclusions

We studied the problem of online rank-one matrix completion in the setting of repeated item recommendations and blocked item recommendations, which should be applicable for several practical recommendation systems. We analyzed an explore-then-commit (ETC) style method which is able to get the regret averaged over users to be nearly independent of number of items. That is, per user, we require only logarithmic many item recommendations to get non-trivial regret bound. But, the dependence on the number of rounds T is sub-optimal. We further improved this dependence by proposing OCTAL that carefully combines exploration, exploitation and clustering for different users/items. Our methods iteratively refines estimate of the underlying reward matrix, while also identifying users which can be recommended certain items confidently. Our algorithms and proof techniques are significantly different than existing bandit learning literature. We believe that our work only scratches the surface of an important problem domain with several open problems. For example, Algorithm 3 requires rank-1 reward matrix. Generalizing the result to rank-r reward matrices would be interesting. Furthermore, relaxing assumptions on the reward matrix like stochastic noise or additive noise should be relevant for several important settings. Finally, collaborative filtering can feed users related items, hence might exacerbate their biases. Our method might actually help mitigate the bias due to explicit exploration, but further investigation into such challenges is important.

REFERENCES

Emmanuel Abbe, Jianqing Fan, Kaizheng Wang, and Yiqiao Zhong. Entrywise eigenvector analysis of random matrices with low expected rank. *Annals of statistics*, 48(3):1452, 2020.

Naman Agarwal, Alon Gonen, and Elad Hazan. Learning in non-convex games with an optimization oracle. In *Conference on Learning Theory*, pp. 18–29. PMLR, 2019.

Junhyung Ahn, Adel Elmahdy, Soheil Mohajer, and Changho Suh. On the fundamental limits of matrix completion: Leveraging hierarchical similarity graphs. *arXiv preprint arXiv:2109.05408*, 2021.

Kwangjun Ahn, Kangwook Lee, Hyunseung Cha, and Changho Suh. Binary rating estimation with graph side information. *Advances in neural information processing systems*, 31, 2018.

Guy Bresler and Mina Karzand. Regret bounds and regimes of optimality for user-user and item-item collaborative filtering. *arXiv:1711.02198*, 2019.

Guy Bresler, George H. Chen, and Devavrat Shah. A latent source model for online collaborative filtering. In *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'14, pp. 3347–3355, 2014.

Guy Bresler, Devavrat Shah, and Luis Filipe Voloch. Collaborative filtering with low regret. In *Proceedings of the 2016 ACM SIGMETRICS International Conference on Measurement and Modeling of Computer Science*, SIGMETRICS '16, pp. 207–220, New York, NY, USA, 2016. ACM.

Sébastien Bubeck. Introduction to online optimization. *Lecture Notes*, 2, 2011.

Sébastien Bubeck and Nicolo Cesa-Bianchi. Regret analysis of stochastic and nonstochastic multi-armed bandit problems. *arXiv preprint arXiv:1204.5721*, 2012.

Emmanuel J Candès and Benjamin Recht. Exact matrix completion via convex optimization. *Foundations of Computational mathematics*, 9(6):717–772, 2009.

Yuxin Chen, Yuejie Chi, Jianqing Fan, Cong Ma, and Yuling Yan. Noisy matrix completion: Understanding statistical guarantees for convex relaxation via nonconvex optimization. *arXiv preprint arXiv:1902.07698*, 2019.

Yash Deshpande and Andrea Montanari. Linear bandits in high dimension and recommendation systems. In *Allerton Conference on Communication, Control, and Computation*, pp. 1750–1754, October 2012.

Adel Elmahdy, Junhyung Ahn, Changho Suh, and Soheil Mohajer. Matrix completion with hierarchical graph side information. *Advances in neural information processing systems*, 33:9061–9074, 2020.

Ken Goldberg, Theresa Roeder, Dhruv Gupta, and Chris Perkins. Eigentaste: A constant time collaborative filtering algorithm. *information retrieval*, 4(2):133–151, 2001.

Botao Hao, Jie Zhou, Zheng Wen, and Will Wei Sun. Low-rank tensor bandits. *arXiv preprint arXiv:2007.15788*, 2020.

Reinhard Heckel and Kannan Ramchandran. The sample complexity of online one-class collaborative filtering. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, ICML'17, pp. 1452–1460. JMLR.org, 2017.

Baihe Huang, Kaixuan Huang, Sham Kakade, Jason D Lee, Qi Lei, Runzhe Wang, and Jiaqi Yang. Optimal gradient-based algorithms for non-concave bandit optimization. *Advances in Neural Information Processing Systems*, 34:29101–29115, 2021.

Xiao-Yu Huang, Bing Liang, and Wubin Li. Online collaborative filtering with local and global consistency. *Information Sciences*, 506:366–382, 2020.

Wasim Huleihel, Soumyabrata Pal, and Ofer Shayevitz. Learning user preferences in non-stationary environments. In *International Conference on Artificial Intelligence and Statistics*, pp. 1432–1440. PMLR, 2021.

Prateek Jain and Purushottam Kar. Non-convex optimization for machine learning. *arXiv preprint arXiv:1712.07897*, 2017.

Prateek Jain, Praneeth Netrapalli, and Sujay Sanghavi. Low-rank matrix completion using alternating minimization. In *Proceedings of the forty-fifth annual ACM symposium on Theory of computing*, pp. 665–674, 2013.

Changhun Jo and Kangwook Lee. Discrete-valued latent preference matrix estimation with graph side information. In *International Conference on Machine Learning*, pp. 5107–5117. PMLR, 2021.

Kwang-Sung Jun, Rebecca Willett, Stephen Wright, and Robert Nowak. Bilinear bandits with low-rank structure. In *International Conference on Machine Learning*, pp. 3163–3172. PMLR, 2019.

Sumeet Katariya, Branislav Kveton, Csaba Szepesvári, Claire Vernade, and Zheng Wen. Bernoulli rank-1 bandits for click feedback. *arXiv preprint arXiv:1703.06513*, 2017a.

Sumeet Katariya, Branislav Kveton, Csaba Szepesvari, Claire Vernade, and Zheng Wen. Stochastic rank-1 bandits. In *Artificial Intelligence and Statistics*, pp. 392–401. PMLR, 2017b.

Yehuda Koren. Factorization meets the neighborhood: A multifaceted collaborative filtering model. In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '08, pp. 426–434, New York, NY, USA, 2008. ACM.

Branislav Kveton, Csaba Szepesvári, Anup Rao, Zheng Wen, Yasin Abbasi-Yadkori, and S Muthukrishnan. Stochastic low-rank bandits. *arXiv preprint arXiv:1712.04644*, 2017.

Tor Lattimore and Csaba Szepesvári. *Bandit algorithms*. Cambridge University Press, 2020.

Jing Lu, Steven Hoi, and Jialei Wang. Second order online collaborative filtering. In *Asian Conference on Machine Learning*, pp. 325–340. PMLR, 2013.

Yangyi Lu, Amirhossein Meisami, and Ambuj Tewari. Low-rank generalized linear bandit problems. In *International Conference on Artificial Intelligence and Statistics*, pp. 460–468. PMLR, 2021.

Rahul Mazumder, Trevor Hastie, and Robert Tibshirani. Spectral regularization algorithms for learning large incomplete matrices. *The Journal of Machine Learning Research*, 11:2287–2322, 2010.

Sahand Negahban and Martin J Wainwright. Restricted strong convexity and weighted matrix completion: Optimal bounds with noise. *The Journal of Machine Learning Research*, 13(1): 1665–1697, 2012.

Rajat Sen, Karthikeyan Shanmugam, Murat Kocaoglu, Alex Dimakis, and Sanjay Shakkottai. Contextual bandits with latent confounders: An nmf approach. In *Artificial Intelligence and Statistics*, pp. 518–527. PMLR, 2017.

Shai Shalev-Shwartz et al. Online learning and online convex optimization. *Foundations and trends in Machine Learning*, 4(2):107–194, 2011.

Arun Sai Suggala and Praneeth Netrapalli. Online non-convex learning: Following the perturbed leader is optimal. In *Algorithmic Learning Theory*, pp. 845–861. PMLR, 2020.

Cindy Trinh, Emilie Kaufmann, Claire Vernade, and Richard Combes. Solving bernoulli rank-one bandits with unimodal thompson sampling. In *Algorithmic Learning Theory*, pp. 862–889. PMLR, 2020.

Joel A Tropp. User-friendly tail bounds for sums of random matrices. *Foundations of computational mathematics*, 12(4):389–434, 2012.

Miao Xu, Rong Jin, and Zhi-Hua Zhou. Speedup matrix completion with side information: Application to multi-label learning. *Advances in neural information processing systems*, 26, 2013.

Lin Yang, Lei Deng, Mohammad H Hajiesmaili, Cheng Tan, and Wing Shing Wong. An optimal algorithm for online non-convex learning. *Proceedings of the ACM on Measurement and Analysis of Computing Systems*, 2(2):1–25, 2018.

Feng Zhang, Ti Gong, Victor E Lee, Gansen Zhao, and Guangzhi Qu. Simple is beautiful: An online collaborative filtering recommendation solution with higher accuracy. In *Asia-Pacific Web Conference*, pp. 497–508. Springer, 2015.

Qiaosheng Zhang, Geewon Suh, Changho Suh, and Vincent YF Tan. Mc2g: An efficient algorithm for matrix completion with social and item similarity graphs. *IEEE Transactions on Signal Processing*, 2022.

Cai-Nicolas Ziegler, Sean M McNee, Joseph A Konstan, and Georg Lausen. Improving recommendation lists through topic diversification. In *Proceedings of the 14th international conference on World Wide Web*, pp. 22–32, 2005.