Multi-Agent Heterogeneous Stochastic Linear Bandits

Avishek Ghosh$^{\star 1}$(✉), Abishek Sankararaman$^{\star 2}$, and Kannan Ramchandran^{3}

[1] Halıcıoğlu Data Science Institute (HDSI), UC San Diego, USA
[2] AWS AI, Palo Alto, USA
[3] Electrical Engg. and Computer Sciences, UC Berkeley, USA
a2ghosh@ucsd.edu, abisanka@amazon.com, kannanr@eecs.berkeley.edu

Abstract. It[4] has been empirically observed in several recommendation systems, that their performance improve as more people join the system by learning *across heterogeneous users.* In this paper, we seek to theoretically understand this phenomenon by studying the problem of minimizing regret in an N users heterogeneous stochastic linear bandits framework. We study this problem under two models of heterogeneity; *(i)* a personalization framework where no two users are necessarily identical, but are all similar, and and *(ii)* a clustering framework where users are partitioned into groups with users in the same group being identical, but different across groups. In the personalization framework, we introduce a natural algorithm where, the personal bandit instances are initialized with the estimates of the global average model and show that, any agent i whose parameter deviates from the population average by ϵ_i, attains a regret scaling of $\widetilde{O}(\epsilon_i\sqrt{T})$. In the clustered users' setup, we propose a successive refinement algorithm, which for any agent, achieves regret scaling as $\mathcal{O}(\sqrt{T/N})$, if the agent is in a 'well separated' cluster, or scales as $\mathcal{O}(T^{\frac{1}{2}+\varepsilon}/(N)^{\frac{1}{2}-\varepsilon})$ if its cluster is not well separated, where ε is positive and arbitrarily close to 0. Our algorithms enjoy several attractive features of being *problem complexity adaptive and parameter free* —if there is structure such as well separated clusters, or all users are similar to each other, then the regret of every agent goes down with N (collaborative gain). On the other hand, in the worst case, the regret of any user is no worse than that of having individual algorithms per user that does not leverage collaborations.

Keywords: Linear bandits · Personalization · Clustering

1 Introduction

Large scale web recommendation systems have become ubiquitous in the modern day, due to a myriad of applications that use them including online shopping services, video streaming services, news and article recommendations, restaurant recommendations etc, each of which are used by thousands, if not more

[4] *Avishek Ghosh and Abishek Sankararaman contributed equally.

users, across the world. For each user, these systems make repeated decisions under uncertainty, in order to better learn the preference of each individual user and serve them. A unique feature these large platforms have is that of *collaborative learning* —namely applying the learning from one user to improve the performance on another [26]. However, the sequential online setting renders this complex, as two users are seldom identical [39].

We study the problem of multi-user contextual bandits [6], and quantify the gains obtained by collaborative learning under user heterogeneity. We propose two models of user-heterogeneity: (a) personalization framework where no two users are necessarily identical, but are close to the population average, and (b) clustering framework where only users in the same group are identical. Both these models are widely used in practical systems involving a large number of users (ex. [28, 32, 39, 42]). The personalization framework in these systems is natural in many neural network models, wherein users represented by learnt embedding vectors are not identical; nevertheless similar users are embedded nearby [37, 38, 45, 48]. Moreover, user clustering in such systems can be induced from a variety of factors such as affinity to similar interests, age-groups etc [33, 38, 43].

Formally, our model consists of N users, all part of a common platform. The interaction between the agents and platform proceeds in a sequence of rounds. Each round begins with the platform receiving K contexts corresponding to K items from the environment. The platform then recommends an item to each user and receives feedback from them about the item. We posit that associated with user i, is an preference vector θ_i^*, initially unknown to the platform. In any round, the average reward (the feedback) received by agent i for a recommendation of item, is the inner product of θ_i^* with the context vector of the recommended item. The goal of the platform is to maximize the reward collected over a time-horizon of T rounds. Following standard terminology, we henceforth refer to an "arm" and item interchangeably, and thus "recommending item k" is synonymous to "playing arm k". We also use agents and users interchangeably.

Example Application: Our setting is motivated through a caricature of a news recommendation system serving N users and K publishers [27]. Each day, each of the K publishers, publishes a news article, which corresponds to the context vector in our contextual bandit framework. In practice, one can use standard tools to embed articles in vector spaces, where the dimensions correspond to topics such as politics, religion, sports etc ([44]). The user preference indicates the interest of a user, and the reward, being computed as an inner product of the context vector and the user preference, models the observation that the more aligned an article is to a user's interest, the higher the reward.

For both frameworks, we propose *adaptive* algorithms; in the personalization framework, our proposed algorithm, namely Personalized Multi-agent Linear Bandits (PMLB) adapts to the level of common representation across users. In particular, if an agents' preference vector is close to the population average, PMLB exploits that and incurs low regret for this agent due to collaboration. On the other hand if an agent's preference vector is far from the population average, PMLB yields a regret similar to that of OFUL [6] or Linear Bandit algorithms [1] that do not benefit from multi-agent collaboration. In the clustering setup, we propose Successive Clustering of Linear Bandits (SCLB), which is agnostic to the number of clusters, the gap between clusters and the cluster size. Yet SCLB yields regret that depends on these parameters, and is thus adaptive.

2 Main Contributions

2.1 Algorithmic: Problem complexity adaptive and (almost) Parameter-Free

We propose adaptive and parameter free algorithms. Roughly speaking, an algorithm is parameter-free and adaptive, if does not need input about the difficulty of the problem, yet has regret guarantees that scale with the inherent complexity. We show in the two frameworks that, if there is structure, then the regret attained by our algorithms is much lower as they learn across users. Simultaneously, in the worst case, the regret guarantee is no worse than if every agent had its own algorithm without collaborations.

In the personalization framework, we give PMLB, a parameter free algorithm, whose regret adapts to an appropriately defined problem complexity – if the users are similar, then the regret is low due to collaborative learning while, in the worst case, the regret is no worse than that of individual learning. Formally, we define the complexity as the *factor of common representation*, which for agent i is $\epsilon_i := \|\theta_i^* - \frac{1}{N}\sum_{l=1}^{N}\theta_l^*\|$, where $\theta_i^* \in \mathbb{R}^d$ is agent i's representation, and $\frac{1}{N}\sum_{l=1}^{N}\theta_i^*$ is the average representation of N agents. PMLB adapts to ϵ_i gracefully (without knowing it apriori) and yields a regret of $\mathcal{O}(\epsilon_i\sqrt{dT})$. Hence, if the agents share representations, i.e., ϵ_i is small, then PMLB obtains low regret. On the other hand, if ϵ_i is large, say $\mathcal{O}(1)$, the agents do not share a common representation, the regret of PMLB is $\mathcal{O}(\sqrt{dT})$, which matches that obtained by each agent playing OFUL, independently of other agents. Thus, PMLB benefits from collaborative learning and obtains small regret, if the problem structure admits, else the regret matches the baseline strategy of every agent running an independent bandit instance.

The clustering framework considers the scenario when not all users are identical or near identical. In this framework, the large number of users belong to a few types, with users of the same type having identical parameters, but users across types have different parameters. Assuming that all users are near identical in this setting will not lead to good performance as all users can be far from the average. We give a multi-phase, successive refinement based algorithm, SCLB, which is parameter free—specifically no knowledge of cluster separation and number of clusters is needed. SCLB *automatically* identifies whether a given problem instance is ‘hard’ or ‘easy’ and adapts to the corresponding regret. Concretely, SCLB attains per-agent regret $\mathcal{O}(\sqrt{T/N})$, if the agent is in a ‘well separated’ (i.e. ‘easy’) cluster, or $\mathcal{O}(T^{\frac{1}{2}+\varepsilon}/(N)^{\frac{1}{2}-\varepsilon})$ if the agent's cluster is not well separated (i.e., ‘hard’), where ε is positive and arbitrarily close to 0. *This* *result holds true, even in the limit when the cluster separation approaches* 0. This shows that when the underlying instance gets harder to cluster, the regret is increased. Nevertheless, despite the clustering being hard to accomplish, every user still experiences collaborative gain of $N^{1/2-\varepsilon}$ and regret sub-linear in T. Moreover, if clustering is easy i.e., well-separated, then the regret rate *matches that of an oracle that knows the cluster identities.*

Empirical Validation: We empirically verify the theoretical insights on both synthetic and Last.FM real data. We compare with three benchmarks —CLUB [18], SCLUB [29], and a simple baseline where every agent runs an independent bandit model, i.e., no collaboration. We observe that our algorithms have superior performance compared to the benchmarks in a variety of settings.

2.2 Theoretical: Improved bounds for Clustering

It is worth pointing out that SCLB works for *all* ranges of separation, which is starkly different from standard algorithms in bandit clustering ([17, 18, 23]) and statistics ([3, 24]). We now compare our results to CLUB [18], that can be modified to be applicable to our setting (c.f. Section 7) (note that we make *identical assumptions* to that of CLUB). First, CLUB is non-adaptive and its regret guarantees hold only when the clusters are separated. Second, even in the separated setting, the separation (gap) cannot be lower than $\mathcal{O}(1/T^{1/4})$ for CLUB, while it can be as low as $\mathcal{O}(1/T^{\alpha})$, where $\alpha < 1/2$ for SCLB. Moreover, in simulations (Section 7) we observe that SCLB outperforms CLUB in a variety of synthetic and a real data setting.

2.3 Technical Novelty

The key innovations we introduce in the analysis are that of *'shifted OFUL'* and *'perturbed OFUL'* algorithms in the personalization and clustering setup respectively. In the personalization setup, our algorithm first estimates the mean vector $\bar{\theta^*} := \frac{1}{N}\sum_{i=1}^{N} \theta_i^*$ of the population. Subsequently, the algorithm subtracts the effect of the mean and only learns the component $\theta_i^* - \bar{\theta^*}$ by compensating the rewards. Our technical innovation is to show that with high probability, shifting the rewards by any fixed vector can only increase overall regret (Lemma 7). In the clustering setup, our algorithm first runs individual OFUL instances per agent, estimates the parameter, then clusters the agents and treats all agents of a single cluster as one entity. In order to prove that this works even when the cluster separation is small, we need to analyze the behaviour of OFUL where the rewards come from a slightly perturbed model.

3 Related Work

Collaborative gains in multi-user recommendation systems have long been studied in Information retrieval and recommendation systems (ex. [26, 28, 32, 42]). The focus has been in developing effective ideas to help practitioners deploy large scale systems. Empirical studies of recommendation system has seen renewed interest lately due to the integration of deep learning techniques with classical ideas (ex. [9,34,36,37,47,49]). Motivated by the empirical success, we undertake a theoretical approach to quantify collaborative gains achievable in a contextual bandit setting. Contextual bandits has proven to be fruitful in modeling sequential decision making in many applications [5,18,27].

The framework of personalized learning has been exploited in a great detail in representation learning and meta-learning. While [11,21,25,40,41] learn common representation across agents in Reinforcement Learning, [2] uses it for imitation learning. We remark that representation learning is also closely connected to meta-learning [10,15,22], where close but a common initialization is learnt from leveraging non identical but similar representations. Furthermore, in Federated learning, the problem of personalization is a well studied problem [12,13,35].

The paper of [18] is closest to our clustering setup, where in each round, the platform plays an arm for a single randomly chosen user. This model was then subsequently improved by [30] and [29] which all exploit the fact that the users' unknown vectors are clustered. As outlined before, our algorithm obtains a superior performance, both in theory and empirically. For personalization, the recent papers of [46] and [4] are the closest, which posits all users's parameters to be in a common low dimensional subspace. [46] proposes a learning algorithm under this assumption. In contrast, we make no parametric assumptions, and demonstrate an algorithm that achieves collaboration gain, if there is structure, while degrading gracefully to the simple baseline of independent bandit algorithms in the absence of structure.

4 Problem Setup

Users and Arms: Our system consists of N users, interacting with a centralized system (termed as 'center' henceforth) repeatedly over T rounds. At the beginning of each round, environment provides the center with K context vectors corresponding to K arms, and for each user, the center recommends one of the K arms to play. At the end of the round, every user receives a reward for the arm played, which is observed by the center. The K context vectors in round t are denoted by $\beta_t = [\beta_{1,t}, \ldots, \beta_{K,t}] \in \mathbb{R}^{d \times K}$.

User heterogeneity: Each user i, is associated with a preference vector $\theta_i^* \in \mathbb{R}^d$, and the reward user i obtains from playing arm j at time t is is given by $\langle \beta_{j,t}, \theta^* \rangle + \xi_t$. Thus, the structure of the set of user representations $(\theta_i^*)_{i=1}^N$ govern how much benefit from collaboration can be expected. In the rest of the paper, we consider two instantiations of the setup - a clustering framework and the personalization framework.

Stochastic Assumptions: We follow the framework of [1,6] and assume that $(\xi_t)_{t\geq 1}$ and $(\beta_t)_{t\geq 1}$ are random variables. We denote by $\mathcal{F}_{t-1}$, as the sigma algebra generated by all noise random variables upto and including time $t-1$. We denote by $\mathbb{E}_{t-1}(.)$ and $\mathbb{V}_{t-1}(.)$ as the conditional expectation and conditional variance operators respectively with respect to $\mathcal{F}_{t-1}$. We assume that the $(\xi_t)_{t\geq 1}$

are conditionally sub-Gaussian noise with known parameter σ, conditioned on all the arm choices and realized rewards in the system upto and including time $t-1$. Without loss of generality, we assume $\sigma = 1$ throughout. The contexts $\beta_{i,t}$ are assumed to be drawn from a (coordinate-wise)[5] bounded distribution (i.e., in any distribution supported on $[-c, c]^{\otimes d}$ for some constant c) independent of both the past and $\{\beta_{j,t}\}_{j \neq i}$, satisfying

$$\mathbb{E}_{t-1}[\beta_{i,t}] = 0 \qquad \mathbb{E}_{t-1}[\beta_{i,t}\, \beta_{i,t}^\top] \succeq \rho_{\min} I. \tag{1}$$

Moreover, for any fixed $z \in \mathbb{R}^d$, of unity norm, the random variable $(z^\top \beta_{i,t})^2$ is conditionally sub-Gaussian, for all i, with $\mathbb{V}_{t-1}[(z^\top \beta_{i,t})^2)] \leq 4\rho_{\min}$. This means that the conditional mean of the covariance matrix is zero and the conditional covariance matrix is positive definite with minimum eigenvalue at least $\rho_{\min}$.

Furthermore, the conditional variance assumption is crucially required to apply (1) for contexts of (random) bandit arms selected by our learning algorithm (see [18, Lemma 1]). Note this this set of assumptions is not new and the exact set of assumptions were used in [6, 18][6] for online clustering and binary model selection respectively. Furthermore, [16] uses similar assumptions for stochastic linear bandits and [19] uses it for model selection in Reinforcement learning problems with function approximation.

Example of contexts: Contexts, $\beta_{i,t}$, drawn iid from $\mathsf{Unif}[-1/\sqrt{d}, 1/\sqrt{d}]^{\otimes d}$ satisfy the above conditions, with $\rho_{\min} = c_0/d$ (c_0 : constant). The $1/\sqrt{d}$ scaling ensures that the norm is $\mathcal{O}(1)$. Observe that our stochastic assumption also includes the setting where the distribution of contexts over time follows a random process independent of the actions and rewards from the learning algorithm.

Performance Metric: At time t, we denote by $B_{i,t} \in [K]$ to be the arm played by any agent i with preference vector θ_i^*. The corresponding regret, over a time horizon of T is given by $R_i(T) = \sum_{t=1}^T \mathbb{E} \max_{j \in [K]} \langle \theta_i^*, \beta_{j,t} - \beta_{B_{i,t},t} \rangle$.

Throughout, OFUL refers to the linear bandit algorithm of [1], which we use as a blackbox. In particular we use a variant of the OFUL as prescribed in [6][7].

5 Personalization

In this section, we assume that the users' representations $\{\theta_i^*\}_{i=1}^N$ are similar but not necessarily identical. Of course, without any structural similarity among $\{\theta_i^*\}_{i=1}^N$, the only way-out is to learn the parameters separately for each user. In the setup of personalized learning, it is typically assumed that (see [8, 14, 31, 46] and the references therein) that the parameters $\{\theta_i^*\}_{i=1}^N$ share some commonality, and the job is to learn the shared components or representations of $\{\theta_i^*\}_{i=1}^N$

[5] In the clustering framework, we were able to remove this coordinate-wise bounded assumption. We only assume boundedness in ℓ_2 norm.

[6] The conditional variance assumption is implicitly used in [6].

[7] We use OFUL as used in the OSOM algorithm of [6] without bias for the linear contextual setting.

Algorithm 1: Personalized Multi-agent Linear Bandits (PMLB)

```
Input: Agents $N$, Horizon $T$
Common representation learning : Estimate $\bar{\theta}^* = \frac{1}{N}\sum_{i=1}^N \theta_i^*$
Initialize a single instance of OFUL($\delta$), called common OFUL
for times $t \in \{1, \cdots, \sqrt{T}\}$ do
    All agents play the action given by the common OFUL
    Common OFUL's state updated by average of observed rewards at all agents
end for
$\widehat{\theta}^* \leftarrow$ the parameter estimate of Common OFUL at the end of round $\sqrt{T}$
Personal Learning
for agents $i \in \{1, \ldots, N\}$ in parallel do
    Initialize modified ALB-Norm($\delta$) of [20] instance per agent (reproduced in Algorithm 5 in Supplementary Material)
    for times $t \in \{\sqrt{T}+1, \ldots, T\}$ do
        Agents play arm output by their personal copy of ALB-Norm (denoted as $\beta_{b_t^{(i)},t}$) and receive reward $y_t$
        Every agent updates their ALB-Norm state with corrected reward $\tilde{y}_i^{(t)} = y_i^{(t)} - \langle \beta_{b_t^{(i)},t}, \hat{\theta}^* \rangle$
    end for
end for
```

collaboratively. After learning the common part, the individual representations can be learnt locally at each agent.

We assume, that the contexts are drawn iid from $\mathsf{Unif}[-1/\sqrt{d}, 1/\sqrt{d}]^{\otimes d}$. This is for clarity of exposition and concreteness and without loss of generality, our analysis can be extended to any distribution supported on $[-c, c]^{\otimes d}$. Moreover, we relax this assumption in Section 6. We now define the notion of common representation across users. Let $\|\theta_l^*\| \leq 1$ for all $l \in [N]$. We define $\bar{\theta}^* = \frac{1}{N}\sum_{l=1}^N \theta_l^*$ as the average parameter.

Definition 1. *(ϵ common representation) An agent i has ϵ_i common representation across N agents if $\|\theta_i^* - \bar{\theta}^*\| \leq \epsilon_i$, where ϵ_i is defined as the common representation factor.*

The above definition characterizes how far the representation of agent i is from the average representation $\bar{\theta}^*$. Note that since $\|\theta_l^*\| \leq 1$ for all l, we have $\epsilon_i \leq 2$. Furthermore, if ϵ_i is small, one can hope to exploit the common representation across users. On the other hand, if ϵ_i is large (say $\mathcal{O}(1)$), there is no hope to leverage collaboration across agents.

5.1 The PMLB Algorithm

Algorithm 1 has *(i)* a common learning and *(ii)* a personal fine-tuning phase.

Common Representation Learning: In the first phase, PMLB learns the average representation $\bar{\theta}^*$ by recommending the same arm to all users and averaging the obtained rewards. At the end of this phase, the center has the estimate $\hat{\theta}^*$ of the average representation $\bar{\theta}^*$. Since the algorithm aggregates the reward from all N agents, it turns out that the common representation learning phase can be restricted to $\sqrt{T}$ steps.

Personal Fine-tuning In the personal learning phase, the center learns the vector $\theta_i^* - \hat{\theta}^*$, *independently* for every agent. For learning $\theta_i^* - \hat{\theta}^*$, we employ the Adaptive Linear Bandits-norm (`ALB-norm`) algorithm of [20][8]. `ALB-norm` is adaptive, yielding a norm dependent regret, i.e., depends on $\|\theta_i^* - \hat{\theta}^*\|$. The idea here is to exploit the fact that in the common learning phase we have a good estimate of $\bar{\theta}^*$. Hence, if the common representation factor ϵ_i is small, then $\|\theta_i^* - \hat{\theta}^*\|$ is small, and it reflects in the regret expression. In order to estimate the difference, the center *shifts* the reward by the inner product of the estimate $\hat{\theta}^*$. By exploiting the anti-concentration property of Chi-squared distribution along with some standard results from optimization, we show that the regret of the shifted system is worse than the regret of agent i (both in expectation and in high probability)[9].

Without loss of generality, in what follows, we focus on an arbitrary agent belonging to cluster i and characterize the regret. We assume

$$T \geq C\frac{1}{N}\left[\frac{\tau_{\min}(\delta)\rho_{\min}}{d\log(1/\delta)}\right]^{\frac{1}{2\alpha}}, \tau_{\min}(\delta) = \left[\frac{16}{\rho_{\min}^2} + \frac{8}{3\rho_{\min}}\right]\log(\frac{2dT}{\delta}) \tag{2}$$

5.2 Regret Guarantee for PMLB

Theorem 1. *Playing Algorithm 1 with T time and δ, where $T \geq \tau_{\min}^2(\delta)$ (defined in eqn. (2)) and $d \geq C\log(K^2T)$, then the regret of agent i satisfies*

$$R_i(T) \leq \tilde{\mathcal{O}}(\epsilon_i\sqrt{dT} + \ T^{1/4}\sqrt{\frac{d^2}{\rho_{\min}N}})\log^2(1/\delta),$$

with probability at least $1 - c\delta - \frac{1}{\mathrm{poly}(T)}$.

Remark 1. The leading term in regret is $\tilde{\mathcal{O}}(\epsilon_i\sqrt{dT})$. If the common representation factor ϵ_i is small, PMLB exploits that across agents and as a result the regret is small as well.

Remark 2. Moreover, if ϵ_i is big enough, say $\mathcal{O}(1)$, this implies that there is no common representation across users, and hence collaborative learning is meaning less. In this case, the agents learn individually (by running OFUL), and obtain a regret of $\tilde{\mathcal{O}}(\sqrt{dT})$ with high probability. Note that this is being reflected in Theorem 1, as the regret is $\tilde{\mathcal{O}}(\sqrt{dT})$, when $\epsilon_i = \mathcal{O}(1)$.

[8] In Section 9, we modify ALB-Norm. For parameter θ, the original ALB-Norm yields a regret of $\mathcal{O}[(\|\theta\| + 1)d\sqrt{T}]$, while our modified algorithm obtains $\mathcal{O}(\|\theta\|d\sqrt{T})$.

[9] This is intuitive since, otherwise one can find *appropriate shifts* to reduce the regret of OFUL, which contradicts the optimality of OFUL.

Algorithm 2: Successive Clustering of Linear Bandits (SCLB)

```
Input: No. of users $N$, horizon $T$, parameter $\alpha < 1/2$, constant $C$, high probability bound $\delta$
for phases $1 \le j \le \log_2(T)$ do
    Play CMLB ($\gamma = 3/(N2^j)^\alpha$, horizon $T = 2^j$, high probability $\delta/2^j$, cluster-size $p^* = j^{-2}$)
end for
```

The above remarks imply the adaptivity of PMLB. Without knowing the common representation factor ϵ_i, PMLB indeed adapts to it—meaning that yields a regret that depends on ϵ_i. If ϵ_i is small, PMLB leverages common representation learning across agents, otherwise when ϵ_i is large, it yields a performance equivalent to the individual learning. Note that this is intuitive since with high ϵ_i, the agents share no common representation, and so we do not get a regret improvement in this case by exploiting the actions of other agents.

Remark 3. (Lower Bound) When $\epsilon_i = 0$, i.e., in the case when all agents have the identical vectors θ_i^*, then Theorem 1 gives a regret scaling as $R_i(T) \le \widetilde{\mathcal{O}}(T^{1/4} d\sqrt{\frac{1}{\rho_{\min}N}})$. When the contexts are adversarily generated, [7] obtain a lower bound (in expectation) of $\Omega(\sqrt{dT})$. However, in the presence of stochastic context, a lower bound on the contextual bandit problem is unknown to the best of our knowledge.

The requirement on d in Theorem 1 can be removed for expected regret.

Corollary 1. *(Expected Regret) Suppose $T \ge \tau_{\min}^2(\delta)$ for $\delta > 0$. The expected regret of the i-th agent after running Algorithm 1 for T time steps is given by*

$$\mathbb{E}[R_i(T)] \le \tilde{\mathcal{O}}(\epsilon_i \sqrt{dT} + \ T^{1/4} \ \sqrt{\frac{d^2}{\rho_{\min} N}}).$$

6 Clustering

We now propose the clustering framework. Here, we assume that instead of being coordinate-wise bounded, the contexts, $\beta_{i,t} \in \mathbb{B}^d(1)$. The users' vectors $\{\theta_u^*\}_{u=1}^N$ are clustered into L groups, with $p_i \in (0,1]$ denoting the fraction of users in cluster i. All users in the same cluster have the same the preference vector–denoted by θ_i^* for cluster $i \in [L]$. We define *separation parameter*, or SNR (signal to noise ratio) of cluster i as $\Delta_i := \min_{j \in [L] \setminus \{i\}} \|\theta_i^* - \theta_j^*\|$, smallest distance to another cluster.

Learning Algorithm: We propose the Successive Clustering of Linear Bandits (SCLB) algorithm in Algorithm 2. SCLB does not need any knowledge of the gap $\{\Delta_i\}_{i=1}^L$, the number of clusters L or the cluster size fractions $\{p_i\}_{i=1}^L$. Nevertheless, SCLB adapts to the problem SNR and yields regret accordingly. One

Algorithm 3: Clustered Multi-Agent Bandits (CMLB)

```
Input: No. of users $N$, horizon $T$, parameter $\alpha < 1/2$, constant $C$, high probability bound $\delta$, threshold $\gamma$, cluster-size parameter $p^*$
 Individual Learning Phase
$T_{\text{Explore}} \leftarrow C^{(2)} d(NT)^{2\alpha} \log(1/\delta)$
All agents play OFUL($\delta$) independently for $T_{\text{explore}}$ rounds
$\{\hat{\theta}^{(u)}\}_{u=1}^{N} \leftarrow$ All agents' estimates at the end of round $T_{\text{explore}}$.
 Cluster the Users
User-Clusters $\leftarrow$ MAXIMAL-CLUSTER($\{\hat{\theta}^{(u)}\}_{u=1}^{N}, \gamma$, $p^*$)
 Collaborative Learning Phase
Initialize one OFUL($\delta$) instance per-cluster
for clusters $\ell \in \{1, \ldots, |$User-Clusters$|\}$ in parallel do
  for times $t \in \{T_{\text{explore}} + 1, \cdots, T\}$ do
    All users in the $\ell$-th cluster play the arm given by the OFUL algorithm of cluster $l$.
    Average of the observed rewards of all users of cluster $l$ is used to update the OFUL($\delta$) state of cluster $l$
  end for
end for
```

attractive feature of Algorithm 2 is that it works uniformly *for all* ranges of the gap $\{\Delta_i\}_{i=1}^{L}$. This is in sharp contrast with the existing algorithms [18] which is only guaranteed to give good performance when the gap $\{\Delta_i\}_{i=1}^{L}$ are large enough. Furthermore, our uniform guarantees are in contrast with the works in standard clustering algorithms, where theoretical guarantees are only given for a sufficiently large separation [3, 24].

SCLB is a multi-phase algorithm, invoking Clustered Multi-agent Linear Bandits (CMLB) (Algorithm 3) repeatedly, by decreasing the size parameter, namely p^* polynomially and high probability parameter δ_j exponentially. Algorithm 2 proceeds in phases of exponentially growing phase length with phase $j \in \mathbb{N}$ lasting for 2^j rounds. In each phase, a fresh instance of CMLB is instantiated with high probability parameter $\delta/2^j$ and the minimum size parameter j^{-2}. As the phase length grows, the size parameter sent as input to Algorithm 3 decays. This simple strategy suffices to show that the size parameter converges to p_i, and we obtain collaborative gains without knowledge of p_i.

CMLB (Algorithm 3) : CMLB works in the three phases: (a) (Individual Learning) the N users play an independent linear bandit algorithm to (roughly) learn their preference; (b) (Clustering) users are clustered based on their estimates using `MAXIMAL CLUSTER` (Algorithm 4); and (c) (Collaborative Learning) one Linear Bandit instance per cluster is initialized and all users of a cluster play the same arm. The average reward over all users in the cluster is used to update the per-cluster bandit instance. When clustered correctly, the learning is faster, as the noise variance is reduced due to averaging across users. Note that `MAXIMAL CLUSTER` algorithm requires a size parameter p^*.

6.1 Regret guarantee of SCLB

As mentioned earlier, SCLB is an adaptive algorithm that yields provable regret for *all ranges* of $\{\Delta_i\}_{i=1}^L$. When $\{\Delta_i\}_{i=1}^L$ are large, SCLB can cluster the agents perfectly, and thereafter exploit the collaborative gains across users in same cluster. On the other hand, if $\{\Delta_i\}_{i=1}^L$ are small, SCLB still adapts to the gap, and yields a non-trivial (but sub-optimal) regret. As a special case, we show that if all the clusters are very close to one another, then with high probability, SCLB identifies treats all agents as *one big* cluster, yielding highest collaborative gain.

Definition 2 (α-Separable Cluster). *For a fixed $\alpha < 1/2$, cluster $i \in [L]$ is termed α-separable if $\Delta_i \geq \frac{5}{(NT)^\alpha}$. Otherwise, it is termed as α-inseparable.*

Lemma 1. *If CMLB is run with parameters $\gamma = 3/(NT)^\alpha$ and $p^* \leq p_i$ and $\alpha < \frac{1}{2}$, then with probability at least $1 - 2\binom{N}{2}\delta$, any cluster i that is α-separable is clustered correctly. Furthermore, the regret of any user in the α-separated cluster i satisfies,*

$$R_i(T) \leq C_1 \left[\frac{d}{\rho_{\min}}(NT)^\alpha + \sqrt{\frac{d}{\rho_{\min}}}(\sqrt{\frac{T - \frac{d(NT)^{2\alpha}}{\rho_{\min}}\log(1/\delta)}{p_i N}}) \right] \log(1/\delta),$$

with probability exceeding $1 - 4\binom{N}{2}\delta$.

We now present the regret of SCLB for the setting with separable cluster

Theorem 2. *If Algorithm 2 is run for T steps with parameter $\alpha < \frac{1}{2}$, then the regret of any agent in a cluster i that is α-separated satisfies*

$$R_i(T) \leq 4\left(2^{\frac{1}{\sqrt{p_i}}}\right) + C_2\left[\frac{d}{\rho_{\min}}(NT)^\alpha + \sqrt{\frac{dT}{\rho_{\min} N}}\right] \log^2(T)\log(1/\delta),$$

with probability at-least $1 - cN^2\delta$. Moreover, if $\alpha \leq \frac{1}{2}(\frac{\log\left[\frac{\rho_{\min} T}{d p_i N}\right]}{\log(NT)})$, we have $R_i(T) \leq \tilde{\mathcal{O}}[2^{\frac{1}{\sqrt{p_i}}} + \sqrt{\frac{d}{\rho_{\min}}}\ \sqrt{\frac{T}{N}}]\log(1/\delta)$.

Remark 4. Note that we obtain the regret scaling of $\tilde{\mathcal{O}}(\sqrt{T/N})$, which is optimal, i.e., the regret rate matches an oracle that knows cluster membership. The cost of successive clustering is $\mathcal{O}(2^{\frac{1}{\sqrt{p_i}}})$, which is a T-independent (problem dependent) constant.

Remark 5. Note that the separation we need is only $5/(NT)^\alpha$. This is a weak condition since in a collaborative system with large N and T, this quantity is sufficiently small.

```
Algorithm 4: MAXIMAL-CLUSTER
Input: All estimates $\{\hat{\theta}^{(i)}\}_{i=1}^{N}$, size parameter $p^* > 0$, threshold $\gamma \geq 0$.
Construct an undirected Graph $G$ on $N$ vertices as follows:
    $||\widehat{\theta}_i^* - \widehat{\theta}_j^*|| \leq \gamma \Leftrightarrow i \sim_G j$
$\mathcal{C} \leftarrow \{C_1, \cdots, C_k\}$ all the connected components of $G$
$\mathcal{S}(p^*) \leftarrow \{C_j : |C_j| < p^* N\}$ {All Components smaller than $p^* N$}
$C^{(p)} \leftarrow \cup_{C \in \mathcal{S}(p^*)} C$ {Collapse all small components into one}
Return : $\mathcal{C} \setminus \mathcal{S}(p^*) \bigcup C^{(p)}$ {Each connected component larger than $p^* N$ is
    a cluster, and all small components are a single cluster}
```

Remark 6. Observe that $R_i(T)$ is a decreasing function of N. Hence, more users in the system ensures that the regret decreases. This is collaborative gain.

Remark 7. (Comparison with [18]) Note that in a setup where clusters are separated, [18] also yields a regret of $\tilde{\mathcal{O}}(\sqrt{T/N})$. However, the separation between the parameters (gap) for [18] cannot be lower than $\mathcal{O}(1/T^{1/4})$, in order to maintain order-wise optimal regret. On the other hand, we can handle separations of the order $\mathcal{O}(1/T^{\alpha})$, and since $\alpha < 1/2$, this is a strict improvement over [18].

Remark 8. The constant term $\mathcal{O}(2^{\frac{1}{\sqrt{p_i}}})$ can be removed if we have an estimate of the p_i. Here, instead of SCLB, we simply run CMLB with the estimate of p_i and obtain the regret of Lemma 1, without the term $\mathcal{O}(2^{\frac{1}{\sqrt{p_i}}})$.

We now present our results when cluster i is α-inseparable.

Lemma 2. *If CMLB is run with input $\gamma = 3/(NT)^{\alpha}$ and $p^* \leq p_i$ and $\alpha < \frac{1}{2}$, then any user in a cluster i that is α-inseparable satisfies*

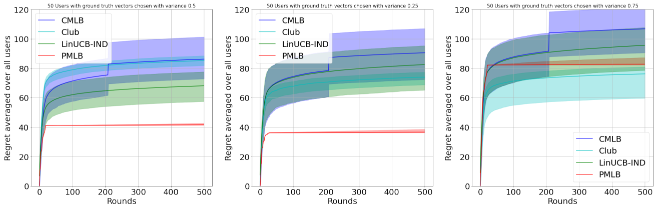
$$R(T) \leq C_1 L(\frac{T^{1-\alpha}}{N^{\alpha}}) + C_2 \sqrt{\frac{d}{\rho_{\min}}}\, [\sqrt{\frac{T - \frac{d(NT)^{2\alpha}}{\rho_{\min}} \log(1/\delta)}{p^* N}}] \log(1/\delta),$$

with probability at least $1 - 4\binom{N}{2}\delta$.

Theorem 3. *If Algorithm 2 is run for T steps with parameter $\alpha < \frac{1}{2}$, then the regret of any agent in a cluster i that is α-inseparable satisfies*

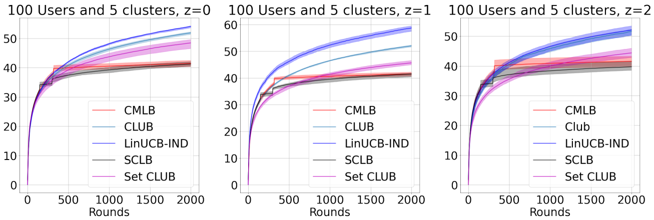
$$R_i(T) \leq 4(2^{\frac{1}{\sqrt{p_i}}}) + C\, L(\frac{T^{1-\alpha}}{N^{\alpha}}) \log(T) + C_1 \sqrt{\frac{dT}{N \rho_{\min}}} \log(1/\delta)\ \log^2(T),$$

with probability at-least $1 - cN^2\delta$. Moreover, if If $\alpha = \frac{1}{2} - \varepsilon$, where ε is a positive constant arbitrarily close to 0, $R(T) \leq \tilde{\mathcal{O}}\left[2^{\frac{1}{\sqrt{p_i}}} + L(\frac{T^{\frac{1}{2}+\varepsilon}}{N^{\frac{1}{2}-\varepsilon}}) + \sqrt{\frac{d}{\rho_{\min}}}\ (\sqrt{\frac{T}{N}}) \log(1/\delta)\right]$.



(a) (b) (c)

Fig. 1: Synthetic simulations of PMLB.



(a) (b) (c)

Fig. 2: Synthetic data simulations for clustering.

Remark 9. As $\varepsilon > 0$, the regret scaling of $\tilde{\mathcal{O}}(\frac{T^{\frac{1}{2}+\varepsilon}}{N^{\frac{1}{2}-\varepsilon}})$ is strictly worse than the optimal rate of $\tilde{\mathcal{O}}(\sqrt{T/N})$. This can be attributed to the fact that the gap (or SNR) can be arbitrarily close to 0, and inseparability of the clusters makes the problem harder to address.

Remark 10. In this setting of low gap (or SNR), where the clusters are inseparable, most existing algorithms (for example [18]) are not applicable. However, we still manage to obtain sub-optimal but non-trivial regret with high probability.

Special case of all clusters being close If $\max_{i\neq j} \|\theta_i^* - \theta_j^*\| \leq 1/(NT)^{\alpha}$, CMLB puts all the users in one big cluster. The collaborative gain in this setting is the largest. Here the regret guarantee of SCLB will be similar to that of Theorem 3 with $p_i = 1$. We defer to Appendix 12 for a detailed analysis.

Remark 11. Observe that if all agents are identical $\max_{i\neq j} \|\theta_i^* - \theta_j^*\| = 0$ our regret bound does not match that of an *oracle* which knows such information. The oracle guarantee would be $\mathcal{O}(\sqrt{T/N})$, whereas our guarantee is strictly worse. The additional regret stems from the universality of our algorithm as it works for all ranges of $\varDelta_i$.

7 Simulations

Personalization setting: In Figure 1, we consider a system where the N ground-truth θ^* vectors are sampled independently from $\mathcal{N}(\mu, \sigma\mathbb{I})$. We choose μ

from the standard normal distribution in each experiment and test performance for different values of σ. Observe that for small σ, all the ground-truth vectors will be close-by (high structure) and when σ is large, the ground-truth vectors are more spread out. We observe in Figure 1 that PMLB adapts to the available structure. With small σ where all users are close to the average, PMLB has much lower regret compared to the baselines. On the other hand, at large σ when there is no structure to exploit, PMLB is comparable to the baselines. This demonstrates empirically that PMLB *adapts* to the problem structure and exploits it whenever present, while not being wore off in the worst case.

Clustering setting : For each plot of Figures 2, users are clustered such that the frequency of cluster i is proportional to i^{-z} (identical to that done in [18]), where z is mentioned in the figures. Thus for $z = 0$, all clusters are balanced, and for larger z, the clusters become imbalanced. For each cluster, the unknown parameter vector θ^* is chosen uniformly at random from the unit sphere. We compare SCLB (ALgorithm 2), CMLB (Algorithm 3) with CLUB [18], Set CLUB [29] and `LinUCB-Ind` the baseline where every agent has an independent copy of OFUL, i.e., no collaboration. (Details in Appendix 17). We observe that our algorithm is competitive with respect to CLUB and Set CLUB, and is superior compared to the baseline where each agent is playing an independent copy of OFUL. In particular, we observe either as the clusters become more imbalanced, or as the number of users increases, SCLB and CMLB have a superior performance compared to CLUB and Set CLUB. Furthermore, since SCLB only clusters users logarithmically many number of times, its runtime is faster compared to CLUB.

8 Conclusion

We consider the problem of leveraging user heterogeneity in a multi-agent stochastic bandit problem under (i) a personalization and, (ii) a clustering framework. In both cases, we give novel adaptive algorithms that, without any knowledge of the underlying instance, provides sub-linear regret guarantees. A natural avenue for future work will be to combine the two frameworks, where users are all not necessarily identical, but at the same time, their preferences are spread out in space (for example the preference vectors are sampled from a Gaussian mixture model). Natural algorithms here will involve first performing a clustering on the population, followed by algorithms such as PMLB. Characterizing performance and demonstrating adaptivity in such settings is left to future work.

References

1. Abbasi-yadkori, Y., Pál, D., Szepesvári, C.: Improved algorithms for linear stochastic bandits. In: Shawe-Taylor, J., Zemel, R., Bartlett, P., Pereira, F., Weinberger, K.Q. (eds.) Advances in Neural Information Processing Systems. vol. 24, pp. 2312–2320. Curran Associates, Inc. (2011)

2. Arora, S., Du, S., Kakade, S., Luo, Y., Saunshi, N.: Provable representation learning for imitation learning via bi-level optimization. In: International Conference on Machine Learning. pp. 367–376. PMLR (2020)
3. Balakrishnan, S., Wainwright, M.J., Yu, B., et al.: Statistical guarantees for the em algorithm: From population to sample-based analysis. Annals of Statistics **45**(1), 77–120 (2017)
4. Ban, Y., He, J.: Local clustering in contextual multi-armed bandits. In: Proceedings of the Web Conference 2021. pp. 2335–2346 (2021)
5. Cesa-Bianchi, N., Gentile, C., Zappella, G.: A gang of bandits. arXiv preprint arXiv:1306.0811 (2013)
6. Chatterji, N., Muthukumar, V., Bartlett, P.: Osom: A simultaneously optimal algorithm for multi-armed and linear contextual bandits. In: International Conference on Artificial Intelligence and Statistics. pp. 1844–1854. PMLR (2020)
7. Chu, W., Li, L., Reyzin, L., Schapire, R.: Contextual bandits with linear payoff functions. In: Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics. pp. 208–214. JMLR Workshop and Conference Proceedings (2011)
8. Collins, L., Hassani, H., Mokhtari, A., Shakkottai, S.: Exploiting shared representations for personalized federated learning. arXiv preprint arXiv:2102.07078 (2021)
9. Covington, P., Adams, J., Sargin, E.: Deep neural networks for youtube recommendations. In: Proceedings of the 10th ACM conference on recommender systems. pp. 191–198 (2016)
10. Denevi, G., Ciliberto, C., Grazzi, R., Pontil, M.: Learning-to-learn stochastic gradient descent with biased regularization. In: Chaudhuri, K., Salakhutdinov, R. (eds.) Proceedings of the 36th International Conference on Machine Learning. Proceedings of Machine Learning Research, vol. 97, pp. 1566–1575. PMLR (09–15 Jun 2019), `http://proceedings.mlr.press/v97/denevi19a.html`
11. D'Eramo, C., Tateo, D., Bonarini, A., Restelli, M., Peters, J.: Sharing knowledge in multi-task deep reinforcement learning. In: International Conference on Learning Representations (2019)
12. Fallah, A., Mokhtari, A., Ozdaglar, A.: On the convergence theory of gradient-based model-agnostic meta-learning algorithms. In: International Conference on Artificial Intelligence and Statistics. pp. 1082–1092. PMLR (2020)
13. Fallah, A., Mokhtari, A., Ozdaglar, A.: Personalized federated learning: A meta-learning approach. arXiv preprint arXiv:2002.07948 (2020)
14. Fallah, A., Mokhtari, A., Ozdaglar, A.: Personalized federated learning with theoretical guarantees: A model-agnostic meta-learning approach. In: Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M.F., Lin, H. (eds.) Advances in Neural Information Processing Systems. vol. 33, pp. 3557–3568. Curran Associates, Inc. (2020), `https://proceedings.neurips.cc/paper/2020/file/24389bfe4fe2eba8bf9aa9203a44cdad-Paper.pdf`
15. Finn, C., Rajeswaran, A., Kakade, S., Levine, S.: Online meta-learning. In: International Conference on Machine Learning. pp. 1920–1930. PMLR (2019)
16. Foster, D.J., Krishnamurthy, A., Luo, H.: Model selection for contextual bandits (2019)
17. Gentile, C., Li, S., Kar, P., Karatzoglou, A., Zappella, G., Etrue, E.: On context-dependent clustering of bandits. In: International Conference on Machine Learning. pp. 1253–1262. PMLR (2017)
18. Gentile, C., Li, S., Zappella, G.: Online clustering of bandits. In: International Conference on Machine Learning. pp. 757–765. PMLR (2014)

19. Ghosh, A., Chowdhury, S.R., Ramchandran, K.: Model selection with near optimal rates for reinforcement learning with general model classes. arXiv preprint arXiv:2107.05849 (2021)
20. Ghosh, A., Sankararaman, A., Kannan, R.: Problem-complexity adaptive model selection for stochastic linear bandits. In: Banerjee, A., Fukumizu, K. (eds.) Proceedings of The 24th International Conference on Artificial Intelligence and Statistics. Proceedings of Machine Learning Research, vol. 130, pp. 1396–1404. PMLR (13–15 Apr 2021), `http://proceedings.mlr.press/v130/ghosh21a.html`
21. Higgins, I., Pal, A., Rusu, A., Matthey, L., Burgess, C., Pritzel, A., Botvinick, M., Blundell, C., Lerchner, A.: Darla: Improving zero-shot transfer in reinforcement learning. In: International Conference on Machine Learning. pp. 1480–1490. PMLR (2017)
22. Khodak, M., Balcan, M.F., Talwalkar, A.: Adaptive gradient-based meta-learning methods. arXiv preprint arXiv:1906.02717 (2019)
23. Korda, N., Szorenyi, B., Li, S.: Distributed clustering of linear bandits in peer to peer networks. In: International conference on machine learning. pp. 1301–1309. PMLR (2016)
24. Kwon, J., Caramanis, C.: The em algorithm gives sample-optimality for learning mixtures of well-separated gaussians. In: Conference on Learning Theory. pp. 2425–2487. PMLR (2020)
25. Lazaric, A., Restelli, M.: Transfer from multiple mdps. In: Shawe-Taylor, J., Zemel, R., Bartlett, P., Pereira, F., Weinberger, K.Q. (eds.) Advances in Neural Information Processing Systems. vol. 24. Curran Associates, Inc. (2011), `https://proceedings.neurips.cc/paper/2011/file/fe7ee8fc1959cc7214fa21c4840dff0a-Paper.pdf`
26. Lee, W.S.: Collaborative learning for recommender systems. In: ICML. vol. 1, pp. 314–321. Citeseer (2001)
27. Li, L., Chu, W., Langford, J., Schapire, R.E.: A contextual-bandit approach to personalized news article recommendation. In: Proceedings of the 19th international conference on World wide web. pp. 661–670 (2010)
28. Li, Q., Kim, B.M.: Clustering approach for hybrid recommender system. In: Proceedings IEEE/WIC International Conference on Web Intelligence (WI 2003). pp. 33–38. IEEE (2003)
29. Li, S., Chen, W., Leung, K.S.: Improved algorithm on online clustering of bandits. arXiv preprint arXiv:1902.09162 (2019)
30. Li, S., Karatzoglou, A., Gentile, C.: Collaborative filtering bandits. In: Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval. pp. 539–548 (2016)
31. Li, T., Hu, S., Beirami, A., Smith, V.: Federated multi-task learning for competing constraints. CoRR **abs/2012.04221** (2020), `https://arxiv.org/abs/2012.04221`
32. Linden, G., Smith, B., York, J.: Amazon. com recommendations: Item-to-item collaborative filtering. IEEE Internet computing **7**(1), 76–80 (2003)
33. Liu, Y., Liu, Z., Chua, T.S., Sun, M.: Topical word embeddings. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 29 (2015)
34. Ma, Y., Narayanaswamy, B., Lin, H., Ding, H.: Temporal-contextual recommendation in real-time. In: Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. pp. 2291–2299 (2020)
35. Mansour, Y., Mohri, M., Ro, J., Suresh, A.T.: Three approaches for personalization with applications to federated learning. CoRR **abs/2002.10619** (2020), `https://arxiv.org/abs/2002.10619`

36. Naumov, M., Mudigere, D., Shi, H.J.M., Huang, J., Sundaraman, N., Park, J., Wang, X., Gupta, U., Wu, C.J., Azzolini, A.G., et al.: Deep learning recommendation model for personalization and recommendation systems. arXiv preprint arXiv:1906.00091 (2019)
37. Okura, S., Tagami, Y., Ono, S., Tajima, A.: Embedding-based news recommendation for millions of users. In: Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. pp. 1933–1942 (2017)
38. Ozsoy, M.G.: From word embeddings to item recommendation. arXiv preprint arXiv:1601.01356 (2016)
39. Pal, A., Eksombatchai, C., Zhou, Y., Zhao, B., Rosenberg, C., Leskovec, J.: Pinnersage: Multi-modal user embedding framework for recommendations at pinterest. In: Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. pp. 2311–2320 (2020)
40. Parisotto, E., Ba, J.L., Salakhutdinov, R.: Actor-mimic: Deep multitask and transfer reinforcement learning. arXiv preprint arXiv:1511.06342 (2015)
41. Rusu, A.A., Colmenarejo, S.G., Gulcehre, C., Desjardins, G., Kirkpatrick, J., Pascanu, R., Mnih, V., Kavukcuoglu, K., Hadsell, R.: Policy distillation. arXiv preprint arXiv:1511.06295 (2015)
42. Sarwar, B.M., Karypis, G., Konstan, J., Riedl, J.: Recommender systems for large-scale e-commerce: Scalable neighborhood formation using clustering. In: Proceedings of the fifth international conference on computer and information technology. vol. 1, pp. 291–324. Citeseer (2002)
43. Saveski, M., Mantrach, A.: Item cold-start recommendations: learning local collective embeddings. In: Proceedings of the 8th ACM Conference on Recommender systems. pp. 89–96 (2014)
44. Wang, S., Tang, J., Aggarwal, C., Liu, H.: Linked document embedding for classification. In: Proceedings of the 25th ACM international on conference on information and knowledge management. pp. 115–124 (2016)
45. Xue, H.J., Dai, X., Zhang, J., Huang, S., Chen, J.: Deep matrix factorization models for recommender systems. In: IJCAI. vol. 17, pp. 3203–3209. Melbourne, Australia (2017)
46. Yang, J., Hu, W., Lee, J.D., Du, S.S.: Impact of representation learning in linear bandits. In: International Conference on Learning Representations (2021), `https://openreview.net/forum?id=edJ_HipawCa`
47. Yao, T., Yi, X., Cheng, D.Z., Yu, F., Menon, A., Hong, L., Chi, E.H., Tjoa, S., Ettinger, E., et al.: Self-supervised learning for deep models in recommendations. arXiv preprint arXiv:2007.12865 (2020)
48. Zhao, H., Ding, Z., Fu, Y.: Multi-view clustering via deep matrix factorization. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 31 (2017)
49. Zhao, Z., Hong, L., Wei, L., Chen, J., Nath, A., Andrews, S., Kumthekar, A., Sathiamoorthy, M., Yi, X., Chi, E.: Recommending what video to watch next: a multitask ranking system. In: Proceedings of the 13th ACM Conference on Recommender Systems. pp. 43–51 (2019)