ECAI 2023
K. Gal et al. (Eds.)

doi:10.3233/FAIA230271

Cooperative Thresholded Lasso for Sparse Linear Bandit

Hanieyh Barghi, Xiaotong Cheng and Setareh Maghsudi

Eberhard Karls University of Tübingen, Tübingen, Germany
Hanieyh Barghi https://orcid.org/0009-0009-2353-1270

Abstract. We present a novel approach to address the multi-agent sparse contextual linear bandit problem, in which the feature vectors have a high dimension d whereas the reward function depends on only a limited set of features - precisely $s_0 \ll d$. Furthermore, the learning follows under information-sharing constraints. The proposed method employs Lasso regression for dimension reduction, allowing each agent to independently estimate an approximate set of main dimensions and share that information with others depending on the network's structure. The information is then aggregated through a specific process and shared with all agents. Each agent then resolves the problem with ridge regression focusing solely on the extracted dimensions. We represent algorithms for both a star-shaped network and a peer-to-peer network. The approaches effectively reduce communication costs while ensuring minimal cumulative regret per agent. Theoretically, we show that our proposed methods have a regret bound of order $\mathcal{O}(s_0 \log d + s_0\sqrt{T})$ with high probability, where T is the time horizon. To our best knowledge, it is the first algorithm that tackles row-wise distributed data in sparse linear bandits, achieving comparable performance compared to the state-of-the-art single and multi-agent methods. Besides, it is widely applicable to high-dimensional multi-agent problems where efficient feature extraction is critical for minimizing regret. To validate the effectiveness of our approach, we present experimental results on both synthetic and real-world datasets.

1 Introduction

Cooperative multi-agent bandit is a suitable framework to tackle complex decision-making problems across a broad spectrum of applications such as Ad-Hoc networks [22], personalized recommendation systems [14], traffic management [31], and the like. In such a framework, the challenge is to enable each agent to learn from its own experiences while considering the actions and rewards of other agents in the system. Given the limitations imposed by the environment, it is crucial to simultaneously keep communication between agents to a minimum during the learning process. To simplify the bandit problems with a large set of arms, it is common to assume a specific model for the payoff functions [26], e.g., the linear structure between actions and rewards [21].

The state-of-the-art research about multi-agent linear bandit problems seldom considers the high dimensional action space [11, 13]. The dimension of the action space accounts for a dominant part in both regret bound [17] and communication cost [9, 18]. Real-world settings often entail noisy components comprising web or mobile-based contexts [17, 6], while most relevant features are small and yield a sparse model parameter. The main challenge in sparse linear bandits is learning the sparse structure of the reward function, as only a small subset of features are relevant for prediction, whereas others are irrelevant or noisy. By relying on prior knowledge or presumptions about the sparsity structure, the sparse linear bandit framework offers a potent mathematical model to address this challenge [27, 2, 24].

We propose a novel multi-agent linear bandit algorithm that handles high-dimensional action space when only a minor subset of dimensions is related to the reward. Despite its versatility, a multi-agent version of sparse linear bandits remains unexplored. We develop a collaborative information-sharing mechanism with low communication cost that assists the agent in fast and accurate estimation of the support set of the sparse parameter. To the best of our knowledge, only [9] tackles decentralized sparse bandits; Nevertheless, compared to our model, it includes several limiting assumptions. Specifically, our cooperative framework integrates information sharing among agents into the high-dimensional linear bandit algorithm. Besides, it does not require any prior knowledge regarding the sparse structure. Our main contributions are summarized as follows.

- **The CTL Algorithm:** We propose an innovative solution, namely, the Cooperative Thresholded Lasso Linear bandit (CTL) algorithm, for the multi-agent sparse linear bandit problem. Our proposal leverages the combination of ridge estimation for arm selection and parameter estimation and thresholded Lasso bandit for dimension reduction. We consider two variants of in-network communication: i) centralized framework, where a central server node aggregates the information from all agents and then distributes the results to them, and ii) decentralized peer-to-peer framework, where agents communicate directly with each other without coordination of the central server node. To reduce the communication burden, we propose a communication framework that reduces the total communication rounds to $\mathcal{O}(\log T)$. Our algorithm is simple and easily generalizable, meaning that it can accommodate other dimension reduction techniques or different communication network topologies using little adaptations. That remarkable robustness and flexibility make CTL an attractive solution for multi-agent sparse linear bandit problems.
- **Performance Evaluation:** We establish that the high probability group regret bound of our proposed CTL algorithm is $\mathcal{O}(s_0 \log d + s_0\sqrt{T})$, where T, s_0 and d refer to the number of time steps, non-zero elements in the feature vector and the dimension of the feature vector, respectively. Besides, we prove that the total communication cost is $\mathcal{O}(s_0 \log T)$. These bounds show that CTL is a practical solution for multi-agent sparse linear bandit problems that balances the trade-off between communication and computation cost while retaining low cumulative regret.
- **Numerical Experiment:** We demonstrate the efficacy of the CTL

algorithm through extensive numerical experiments. We compare our proposed algorithm with a series of state-of-the-art sparse linear bandit algorithms, including the Thresholded Lasso [4], Sparsity-Agnostic Lasso [24], and Doubly-Robust Lasso [17]. Experiments on synthetic- and real-world datasets show the superior performance of our proposal. The results highlight the advantages of utilizing a multi-agent framework compared to a single agent with the same number of observations. Besides, we compare our method with a multi-agent low-dimensional algorithm [9], effectively showing the superiority of our approach. The CTL algorithm outperforms the referenced method not only in terms of cumulative regret but also by imposing significantly fewer simplification assumptions.

1.1 Related Works

Our work is closely related to the research on multi-agent linear bandits and sparse linear bandits. In this section, we review the state-of-the-art research in both directions and then highlight connections between them.

Multi-agent bandit problem has gained great attention in the past few years [30, 3, 12]. The proposed strategies for multi-agent problems are dividable into two main categories. Most related works assume that agents continually select from a limited subset of arms and exchange their beliefs about the best arm in their playing set [7, 8, 28]. Reference [7] proposes a teamwork model in which the agents decide whether to pull the arms of a bandit or broadcast their obtained rewards over several epochs, aiming to maximize the total rewards. The model captures a three-way tradeoff between exploration, exploitation, and communication. They also show that the proposed decentralized algorithm with a Value-of-Information communication strategy converges rapidly to the performance of a centralized method. However, in our research and some others, all agents face the same environment; That is, they share the entire set of arms among agents. Thus, the algorithm must consider all arms at each time step. Our research is tightly related to the literature in multi-agent linear bandits such as [29, 18]. Wang et al. [29] present a communication-efficient algorithm under the coordination of a central server, allowing every agent to have immediate access to the complete full network information. Compared to [29], our proposed algorithm considers both a centralized and decentralized structure; Hence it covers a wide range of applications. Reference [18] studies distributed linear bandits in peer-to-peer networks, where each agent can only send information to one randomly chosen agent per round. We consider centralized and decentralized communication networks and allow for less frequent communications. Besides, to our best knowledge, previous research rarely considers the sparse parameter.

Another line of research related to ours is the sparse linear bandit problem. Some papers in that direction assume the availability of side information about the sparse model parameters [2, 15, 17]. For example, to tackle the sparse linear stochastic bandit problem, [2] introduces a technique, namely, online-to-confidence-set conversion, to construct high-probability confidence sets for linear prediction with correlated inputs. However, it requires the sparsity level of the model, i.e., the size of the support set. Reference [15] leverages ideas from linear Thompson sampling and relevance vector machines, resulting in a scalable approach that adapts to the unknown sparse support. That paper also assumes prior knowledge of a slightly larger set of support for the model parameter. Recently, studies on sparse linear bandits overcome that limitation [24, 4], which do not require any prior information about the sparse parameter of the model. Moreover, thresholding has become a natural and efficient way to feature selection in online and offline learning [4, 32, 25], which achieves excellent performance in sparse linear bandit. Thus, establishing a dependable interval for the threshold value is necessary and crucial for the algorithm to operate effectively. The most proximate study to our current context can be found in [4], wherein the authors employ the Lasso framework coupled with thresholding to retain and revise the support set of the model parameter. However, our endeavor not only operates within the domain of a multi-agent scenario but also presents a contrast in the feature selection process. Unlike the approach described in [4], where the feature selection occurs at every time step, our method accomplishes it only logarithmically with respect to T, total time steps, leading to significant computational advantages without compromising the overall accuracy of the results. As a result, the computational expense associated with Lasso estimation diminishes significantly while simultaneously upholding the overall accuracy of the final outcome.

The paper structure is as follows: Firstly, in Section 2, we provide a formal statement of the problem. Then, we discuss the algorithm for centralized and peer-to-peer settings in Section 3. In Section 4, we establish a regret bound for our proposed algorithm. We evaluate our proposal numerical using synthetic- and real datasets in Section 5.

2 Problem Formulation

2.1 Model and Notation

We consider a multi-agent high-dimensional linear bandit problem with N agents. Let T become the problem horizon, i.e., the number of rounds to be played. At each time step $t \in [T]$, each agent $i \in [N]$ receives a set of K context vectors $\mathcal{A}_t^i \subset \mathbb{R}^{K\times d}$ sampled from one unknown distribution. Each agent i selects an action $A_t^i \in \mathcal{A}_t^i$ based on the previous observations in round t and obtains the reward y_t^i,

$$y_t^i := \langle A_t^i, \theta^* \rangle + \omega_t^i, \tag{1}$$

where $\theta^* \in \mathbb{R}^d$ is an unidentified sparse parameter, and ω_t^i is sub-Gaussian noise with a zero mean. Parameter θ^* and A_t^i are both high-dimensional $d \gg 1$, while parameter θ^* is sparse, which means the number of non-zero elements $s_0 = \|\theta^*\|_0 \ll d$. In other words, θ^* is s_0-sparse and s_0 is a constant but unknown integer. Furthermore, if $\mathcal{F}_t^i$ is the σ-algebra generated by random variables $(\mathcal{A}_1^i, A_1^i, y_1^i, \ldots, A_{t-1}^i, y_{t-1}^i, \mathcal{A}_t^i)$, A_t^i is $\mathcal{F}_t^i$-measurable. The noise term ω_t^i is independent across agents given $\mathcal{F}_t^i$ and A_t^i. Moreover, we have the sub-Gaussian property such that $\mathbb{E}[e^{\alpha\omega_t^i}] \leq e^{\alpha^2\sigma^2/2}, \forall\alpha \in \mathbb{R}$, where σ is a positive constant. This inequality implies that the moment-generating function of ω_t^i exists and is bounded, which is a desirable property in many statistical and mathematical models.

At time step t, the instantaneous expected regret of each agent $i \in [N]$ yields

$$r_t^i := \mathbb{E}[\max_{A\in\mathcal{A}_t^i} \langle A - A_t^i, \theta^* \rangle].$$

The cumulative regret for any agent i is $R_i(T) := \sum_{t=1}^T r_t^i$. The objective of each agent is to minimize its overall cumulative regret as an individual.

Notation The ℓ_0-norm of a vector $x \in \mathbb{R}^d$ is $\|x\|_0 = \sum_{j=1}^d \mathbb{1}\{x_j \neq 0\}$. The set $S(x) := \{j \in [d] = \{1, 2, \ldots, d\} : x_j \neq 0\}$ stands for the support of a vector x. For each agent

i, the empirical Gram matrix that the arms produced under a certain algorithm is represented by $\hat{\Sigma}_{t,i} = \frac{1}{t}\sum_{s=1}^{t} A_{s,i}A_{s,i}^\top$. For any $B \subset [d]$, we define $x_B := (x_{1,B}, \ldots, x_{d,B})^\top$ where for all $j \in [d]$, $x_{j,B} := x_j \mathbb{1}\{j \in B\}$. Additionally, we define $x_{\min}$ as $|x_j|$'s minimal value on its support: $x_{\min} := \min_{j\in S(x)} |x_j|$. The weighted norm-2 of vector $x \in \mathbb{R}^d$ is defined as $\|x\|_A := \sqrt{x^\top A x}$, where $A \in \mathbb{R}^{d\times d}$ is a positive definite matrix. We define the minimum eigenvalue of a matrix A as $\lambda_{\min}(A)$.

2.2 Assumptions

Below, we outline our assumptions that mostly stem from [24, 4], and compare them to those in the related literature.

Assumption 1 (Context vector and parameter constraints) *For the feature vector θ^*, we assume that $\|\theta^*\|_1 \leq s_1$ for some unknown constant s_1 and $\|\theta^*\|_2 \leq s_2$, where s_2 is a positive constant. Besides, we assume that the context vector's ℓ_∞-norm is bounded: for all t, $i \in [N]$ and for all $A \in \mathcal{A}_t^i$, $\|A\|_\infty \leq s_A$, where $s_A > 0$ is a constant.*

Bounded norms of model parameter and feature vectors are common assumptions in high dimensional linear models [17, 20].

Assumption 2 (Compatibility condition) *We specify the compatibility constant $\phi(M, S_0)$ as*

$$\phi^2(M,S_0) := \min_{x:\|x_{S_0}\|_1 \neq 0} \left\{ \frac{s_0 x^\top M x}{\|x_{S_0}\|_1^2} : \|x_{S_0^c}\|_1 \leq 3\|x_{S_0}\|_1 \right\}$$

for a matrix $M \in \mathbb{R}^{d\times d}$ and a set $S_0 \subset [d]$. We assume that for the Gram matrix of the action set $\Sigma := \frac{1}{K}\sum_{k=1}^{K} \mathbb{E}_{\mathcal{A}\sim p_A}\left[A_k A_k^\top\right]$ satisfies $\phi^2(\Sigma, S(\theta^*)) \geq \phi_0^2$, where ϕ_0 is some positive constant.

In the high dimensional statistics literature, the compatibility condition appeared for the first time in [5]. It is similar to the standard Gram matrix positive-definiteness for the ordinary least square estimator for linear models, but less constricting. The compatibility condition ensues that the parameter's truly active components are not strongly correlated. According to many pertinent studies, Assumption 2 is essential for the consistency of the Lasso estimation.

Assumption 3 (Relaxed symmetry [24]) *For the distribution p_A of $\mathcal{A}$, there exists a constant $\nu \geq 1$ such that for all $\vec{A} \in \mathbb{R}^{K\times d}$ with $p_A(\vec{A}) > 0$, $\frac{p_A(\vec{A})}{p_A(-\vec{A})} \leq \nu$.*

Assumption 3 stems from [24]. According to this assumption, the joint distribution p_A may exhibit skewness, but this skewness is subject to some constraints. It is known that a broad class of continuous and discrete distributions, such as Gaussian distributions, multi-dimensional uniform distributions, and Rademacher distributions, satisfy the property of relaxed symmetry. This property ensures that the distribution remains symmetric even in the presence of small deviations from the perfect symmetry, allowing for some degree of skewness while still maintaining overall balance.

Assumption 4 (Balanced covariance [24]) *For any permutation γ of $[K]$, for any integer $k \in \{2, \ldots, K-1\}$ and a fixed θ^*, there exists a constant $C_b > 1$ such that*

$$\begin{aligned} C_b \mathbb{E}_{\mathcal{A}\sim p_A}\big[&(A_{\gamma(1)}A_{\gamma(1)}^\top + A_{\gamma(K)}A_{\gamma(K)}^\top) \\ &\cdot \mathbb{1}\{\langle A_{\gamma(1)}, \theta^*\rangle < \ldots < \langle A_{\gamma(K)}, \theta^*\rangle\}\big] \\ \succeq \mathbb{E}_{\mathcal{A}\sim p_A}&\left[A_{\gamma(k)}A_{\gamma(k)}^\top \mathbb{1}\{\langle A_{\gamma(1)}, \theta^*\rangle < \ldots < \langle A_{\gamma(K)}, \theta^*\rangle\}\right]. \end{aligned}$$

We adapted Assumption 4 from [24]. The statement is valid for a variety of distributions, such as multivariate Gaussian distribution and uniform distribution on the sphere. It still applies when contexts are independent of one another with any arbitrary distributions [24].

Assumption 5 (Sparse positive definiteness) *For each $B \subset [d]$, define $\Sigma_B = \frac{1}{K}\sum_{k=1}^{K} \mathbb{E}_{\mathcal{A}\sim p_A}[A_k(B)A_k(B)^\top]$, where $A_k(B)$ is a $|B|$-dimensional vector, which is extracted from the elements of A_k with indices in B. There exists a positive constant $\alpha > 0$ such that $\forall B \subset [d]$,*

$$\begin{aligned} &|B| \leq s_0 + (4\nu C_b s_0)/\phi_0^2 \quad \text{and} \quad S(\theta^*) \subset B \\ &\Rightarrow \min_{v\in\mathbb{R}^{|B|}:\|v\|_2=1} v^\top \Sigma_B v \geq \alpha. \end{aligned}$$

The parameters ϕ_0, ν, and C_b match those of Assumption 2, 3, and 4. According to Assumption 5, the context distribution around the support of θ^* is sufficiently diverse. In low dimensional linear bandit literature, Assumption 5 is commonly used (e.g., [19, 10, 16]).

Algorithm 1: Centralized Cooperative Thresholded Lasso Bandit Algorithm (CCTL)

```
initialisation: $\lambda_0, \xi, \hat{S}_1 = \{1, \ldots, d\}$, and
    $\forall i \in [N] : M_1^i = I_{d\times d}, b_1^i = 0_{1\times d}$
for $t = 1, 2, \ldots, T$ do
    for agent $i \in [N]$ do
        $\hat{\theta}_t^i \leftarrow (M_t^i)^{-1} b_t^i$
        Observe context vectors of all arms $\mathcal{A}_t^i \in \mathbb{R}^{K\times d}$
        $\tilde{\mathcal{A}}_t^i \leftarrow$ remove dimensions $[d] \setminus \hat{S}_t$ from $\mathcal{A}_t^i$
        Select $k' = \arg\max_{k\in[K]} \langle \tilde{\mathcal{A}}_{t,k}^i, \hat{\theta}_t^i \rangle$
        Observe reward $y_t^i$
        Add $\mathcal{A}_{t,k'}^i$ to $A_i$ and $y_t^i$ to $Y_i$
        Update weights $M_{t+1}^i = M_t^i + \tilde{\mathcal{A}}_{t,k'}^i (\tilde{\mathcal{A}}_{t,k'}^i)^\top$ and
            $b_{t+1}^i = b_t^i + y_t^i \tilde{\mathcal{A}}_{t,k'}^i$
    end
    if $\log_\xi(t) \in \mathbb{N}$ then
        $\lambda_t \leftarrow \lambda_0 \sqrt{\frac{2\log t \log d}{t}}$
        $\mathcal{T}_t \leftarrow N \times \lambda_t$
        for agent $i \in [N]$ do
            $\hat{\theta}_t^i \leftarrow \arg\min_\theta \{ \frac{1}{t} \|Y_i - \langle A_i, \theta \rangle\|_2^2 + \lambda_t \|\theta\|_1 \}$
            $\hat{S}_{t+1}^i \leftarrow \{ j \in [d] : |(\hat{\theta}_t^i)_j| > \mathcal{T}_t \}$
        end
        server: $\hat{S}_{t+1} \leftarrow \bigcup_{i=1}^{N} \hat{S}_{t+1}^i$
        each agent $i \in [N]$: Update $M_{t+1}^i$ and $b_{t+1}^i$
            according to $\hat{S}_{t+1}$
    end
    else
        $\hat{S}_{t+1} \leftarrow \hat{S}_t$
    end
end
```

3 Algorithm

In this section, we present the Cooperative Thresholded Lasso bandit algorithm (CTL), which adapts the concept of thresholding in [4] and the LinUCB algorithm [1]. In this method, each agent selects an action based on an estimate of the feature vector θ^* at each time step t. The estimation follows from two main working components, ridge regression, and the thresholded Lasso. Instead of computing the decision-making policy in high-dimensional space d, with the help of thresholded Lasso, the agents decide in a space with a "reduced" number of dimensions, which reduces the computational cost significantly.

In the federated setting, where agents have different actions and estimations, the communication protocol design is critical. In this section, we first consider a centralized communication framework where there exists a centralized server node that coordinates the communication among agents. Under the centralized communication protocol, each agent periodically communicates with the centralized server and synchronizes itself with other agents. We then extend the framework to a decentralized peer-to-peer network setting. Theoretical guarantees are provided for the performance of both structures.

3.1 Centralized Framework with a Server Node

Algorithm 1 summarizes the centralized version of CTL (CCTL), which operates as follows. Initially, each agent i assigns $M_1^i = I_{d\times d}$ and $b_1^i = 0_{1\times d}$ for use in ridge regression. Additionally, $\hat{S}_t$ provides an estimate of the support set of θ^* and is initialized with $\hat{S}_1 = \{1, \ldots, d\}$, including all dimensions. At each step t, every agent chooses an action optimistically based on the estimated $\hat{\theta}_t^i$, while only considering the dimensions provided in $\hat{S}_t$. After receiving the reward, each agent updates its estimate based on ridge regression. During the synchronization step, when $\log_\xi t \in \mathbb{N}$ and $a > 1$, agents obtain an estimate via Lasso, which is used to estimate the support of θ^* with appropriate thresholding computation. We select the regularizer based on the setting in [4]. Unlike [4], here we only perform one threshold procedure. To save communication costs, agents only share their estimate of θ^*'s support. After synchronization, the server node obtains the final estimate of $\hat{S}_t$ by taking the union of the support sets of the shared sets.

Remark 1 *Similar to [23], the algorithm above is generalizable by selecting a random subset of agents in each synchronization step. That approach allows for more flexibility in network coverage, particularly in scenarios where not all agents are consistently online in the system. Additionally, it enables the management of large-scale systems in which many agents are involved, and limited communication capacity is a potential bottleneck. By carefully selecting only a subset of agents to participate in each synchronization round, the algorithm can effectively balance communication demands with the computational and operational capabilities of the system while still maintaining a high degree of accuracy in the estimation of* $\hat{S}_t$.

3.2 Decentralized Peer-to-Peer Framework

In this scenario, each agent communicates directly with its neighbors via a decentralized peer-to-peer protocol. The communication network is modeled by an undirected network $G = (N, E)$, where $e_{i,j} \in E$ if agent i and j can communicate directly, or in other words, i and j are neighbors. Define $\mathcal{N}_i$ as the neighbors of agent i. At each synchronization step, the algorithm proceeds as follows: Once each agent obtains the estimation of θ^*'s support, it randomly selects a neighbor and receives the corresponding support's estimation of that selected neighbour. This additional information is then integrated into the agent's own support estimation through a union operation, enabling the agent to enhance the recall and robustness of its estimate. Here, recall refers to the probability of the main dimensions appearing in the support estimation. All other steps in the algorithm remain similar to those described in the centralized version. Algorithm 2 summarizes the Decentralized peer-to-peer CTL algorithm (DCTL).

Algorithm 2: Decentralized Peer-to-Peer Cooperative Thresholded Lasso Linear Bandit Algorithm (DCTL)

```
initialisation: $\lambda_0, \xi, \hat{S}_1^i = \{1, \ldots, d\}$, and
    $\forall i \in [N] : M_1^i = I_{d\times d}, b_1^i = 0_{1\times d}$
for $t = 1, 2, \ldots, T$ do
    for agent $i \in [N]$ do
        $\hat{\theta}_t^i \leftarrow (M_t^i)^{-1} b_t^i$
        Observe context vectors of all arms $\mathcal{A}_t^i \in \mathbb{R}^{K\times d}$
        $\tilde{\mathcal{A}}_t^i \leftarrow$ remove dimensions $[d] \setminus \hat{S}_t^i$ from $\mathcal{A}_t^i$
        Select $k' = \arg\max_{k\in[K]} \langle \tilde{\mathcal{A}}_{t,k}^i, \hat{\theta}_t^i \rangle$
        Observe reward $y_t^i$
        Add $\mathcal{A}_{t,k'}^i$ to $A_i$ and $y_t^i$ to $Y_i$
        Update weights $M_{t+1}^i = M_t^i + \tilde{\mathcal{A}}_{t,k'}^i (\tilde{\mathcal{A}}_{t,k'}^i)^\top$ and
            $b_{t+1}^i = b_t^i + y_t^i \tilde{\mathcal{A}}_{t,k'}^i$
    end
    if $\log_\xi(t) \in \mathbb{N}$ then
        $\lambda_t \leftarrow \lambda_0 \sqrt{\frac{2\log t \log d}{t}}$
        $\mathcal{T}_t \leftarrow 2 \times \lambda_t$
        for agent $i \in [N]$ do
            $\hat{\theta}_t^i \leftarrow \arg\min_\theta \{\frac{1}{t}\|Y_i - \langle A_i, \theta\rangle\|_2^2 + \lambda_t \|\theta\|_1\}$
            $\hat{S}_{t+1}^i \leftarrow \{j \in [d] : |(\hat{\theta}_t^i)_j| > \mathcal{T}_t\}$
            Select agent $j \in \mathcal{N}_i$ to communicate and obtain
                its estimate $\tilde{S}_{t+1}^j$
            $\hat{S}_{t+1}^i \leftarrow \tilde{S}_{t+1}^i \bigcup \tilde{S}_{t+1}^j$
            Update $M_{t+1}^i$ and $b_{t+1}^i$ according to $\hat{S}_{t+1}^i$
        end
    end
    else
        $\hat{S}_{t+1} \leftarrow \hat{S}_t^i$
    end
end
```

4 Performance Analysis

4.1 Centralized Framework

Theorem 1 *Consider a system consisting of N agents connected by a server node. Every agent uses Algorithm 1 to select arms in each time step. Under Assumption 1-Assumption 5, we can establish the existence of a positive constant c such that* $\lambda_0 = 4\sqrt{c}\sigma s_A$. *Then, for all d* $\geq \exp(4/c)$ *and T* ≥ 2, *with probability at least* $(1-\delta)$, *the following inequality holds:*

$$
\begin{aligned}
R_i(T) \leq 2 s_A s_1 \tau + \frac{8K\sqrt{\xi}}{\sqrt{\xi}-1}(\sqrt{\xi T}-1) \\
\sqrt{\begin{aligned} &\sigma^2 C_a^2 (s_0 + \frac{16 s_0 \nu C_b}{\phi_0})^2 \log^2 T + \\ &\quad (s_2^2 - 2\sigma^2 \log\delta) C_a (s_0 + \frac{16 s_0 \nu C_b}{\phi_0}) \log T \end{aligned}} \\
+ 2K s_A s_1 \left(\frac{1 - T^{1-2N}}{2N-1} + \frac{4}{N C_0^2} + (s_0 + \frac{16 \nu C_b s_0}{\phi_0^2})^2 \right. \\
\left. \frac{40 s_A \nu C_b}{\alpha} \right),
\end{aligned}
$$

where $\tau = \max\left\{\frac{2\log(2d^2)}{C_0^2}, \exp\left(2\log\xi + \frac{2}{c}\right)\right\}$.

Proof 1 (Proof sketch) *We outline the proof of Theorem 1 as follows.*

- ***Performance Analysis of Estimated Support Set:*** *Given that Algorithm 1 iteratively reduces the dimension, the initial step in evaluating the regret bound for our proposed method entails assessing the estimated support set following each synchronization round. To this end, we present the following Lemma to give a tight lower bound for the probability of the existence of $S(\theta^*)$ and the extent of false positive features in this estimation $(\hat{S}_t)$. We prove Lemma 1 in [33] Appendix A.1.*

  Lemma 1 *(Centralized Framework) Assume that, for each agent $i \in [N]$, assumptions 1, 2, 3, and 4 hold. Then for all $t \geq \frac{2\log(2d^2)}{C_0^2}$ with $C_0 := \min\{\frac{1}{2}, \frac{\phi_0^2}{512 s_0 s_A^2 \nu C_b}\}$ the event $\mathcal{E}_t = \{|\hat{S}_t \setminus S(\theta^*)| \leq \frac{16 s_0 \nu C_b}{\phi_0^2}$ and $S(\theta^*) \subset \hat{S}_t\}$ holds true with probability at least*

  $$1 - \left(2\exp\left(-\frac{t'\lambda_{t'}^2}{32\sigma^2 s_A^2} + \log d\right)\right)^N - \exp\left(-\frac{Nt'C_0^2}{2}\right),$$

  where $t' = \xi^{\lfloor \log_\xi t \rfloor}$.

  Lemma 1 extends the support recovery outcome of the Thresholded Lasso Bandit, as stated in [4], to the scenario of multiple agents exchanging information among each other. The reliance on s_0 instead of d is similar to that of the offline result (as Theorem 3.1 of [32]) and the bandit setting illustrated in Lemma 5.4 of [4]. Our thresholding approach, combined with the allowance of agents to share their estimated sets, facilitates a more precise dimension reduction through the learning process, effectively removing the reliance on d for estimation error when t exceeds $2\log(2d^2)/C_0^2$. This, in turn, leads to improved regret bounds as compared to those established in existing literature, such as [24] or [4].
- ***Minimal Eigenvalue of the Empirical Gram Matrix:*** *We introduce the notion of $\hat{\Sigma}_{t,i}$ as the empirical Gram matrix on the estimated support of agent i, up to time step t. This matrix is a fundamental tool to capture the pairwise relationships between estimated survival probabilities at different time points. The desirable property of positive definiteness of $\hat{\Sigma}_{t,i}$ ensures that it is not only invertible but also allows for the utilization of powerful mathematical tools for statistical inference. Our proposed lemma aims to establish the positive definiteness of $\hat{\Sigma}_{t,i}$, even when the underlying data generating process is not i.i.d. Notably, this lemma shares similarities with Lemma 5.6 presented in [4].*

  Lemma 2 *Under Assumptions 1 and 5, for any agent $i \in [N]$ and for all $t \in [T]$, we have:*

  $$\mathbb{P}(\lambda_{min}(\hat{\Sigma}_{t,i}) \geq \frac{\alpha}{4\nu C_b}|\mathcal{E}_t) \geq$$

  $$1 - \exp\left(\log\left(s_0 + \frac{16 s_0 \nu C_b}{\phi_0^2}\right) - \frac{t'\alpha}{20 s_A \nu C_b (s_0 + \frac{16 s_0 \nu C_b}{\phi_0^2})}\right),$$

  where $t' = \xi^{\lfloor \log_\xi t \rfloor}$.

  The proof of this Lemma is similar to that of Lemma 5.6 in [4], albeit with a minor change. Mainly, in the utilization of Lemma F.10 [4], we must modify the upper bound for the size of estimated support set $\hat{S}_t$ to $s_0 + (16 s_0 \nu C_b)/\phi_0^2$, while retaining all other steps unchanged.
- ***Instantaneous Regret Upper Bound:*** *Below, we state a lemma that serves to bound the instantaneous regret for each agent $i \in [N]$. We prove this lemma in Appendix A.2 based on [1].*

  Lemma 3 *For any $t \in [T]$ and each agent $i \in [N]$, with probability at least $1-\delta$ the instantaneous regret $r_t^i = \mathbb{E}[\max_{A \in \mathcal{A}_t^i}\langle A - A_t^i, \theta^*\rangle]$ is upper bounded as*

  $$r_t^i \leq \sum_{k=1}^{K} \mathbb{E}\left[(\left\|A_{t,k}^i\right\|_{(M_t^i)^{-1}} + \left\|A_t^i\right\|_{(M_t^i)^{-1}})(\sigma\sqrt{\log\left(\frac{\det(M_t^i)}{\delta^2}\right)} + \|\theta\|_2^*)|\mathcal{A}_t^i \in \mathcal{R}_k^i, \mathcal{E}_t, \mathcal{G}_{t,i}^{\frac{\alpha}{4\nu C_b}}\right] + 2K s_A s_1 \left(\mathbb{P}((\mathcal{E}_t)^c) + \mathbb{P}((\mathcal{G}_{t,i}^{\frac{\alpha}{4\nu C_b}})^c|\mathcal{E}_t)\right),$$

  where $\mathcal{R}_k^i := \{\mathcal{A}_t^i \in \mathbb{R}^{K\times d} : k \in \arg\max_{k'}\langle A_{t,k'}^i, \theta^*\rangle\}$ and $\mathcal{G}_{t,i}^{\lambda} := \{\lambda_{\min}(\hat{\Sigma}_{\hat{S}_t}^i) \geq \lambda\}$.

With the aforementioned lemmas, we can prove Theorem 1, as provided in [33] Appendix B.

4.2 Decentralized Peer-to-Peer Framework

Theorem 2 *Consider a network of N agents connected via a fix connected graph. In each time step, the system chooses arms using the algorithm 2. There is a positive constant c such that $\lambda_0 = 4\sqrt{c}\sigma s_A$ under the necessary conditions of 1-5. We hereby declare that the following inequality holds true with a probability of at least $1-\delta$ for any d $\geq \exp(4/c)$ and for all $T \geq 2$*

$$R_i(T) \leq 2 s_A s_1 \tau + \frac{8K\sqrt{\xi}}{\sqrt{\xi}-1}(\sqrt{\xi T}-1) \sqrt{\sigma^2 C_a^2 (s_0 + \frac{16 s_0 \nu C_b}{\phi_0})^2 \log^2 T + (s_2^2 - 2\sigma^2 \log\delta) C_a (s_0 + \frac{16 s_0 \nu C_b}{\phi_0}) \log T} + 2K s_A s_1 \left(\frac{1 - T^{1-2N}}{2N-1} + \frac{4}{NC_0^2} + (s_0 + \frac{16 \nu C_b s_0}{\phi_0^2})^2 \frac{40 s_A \nu C_b}{\alpha}\right),$$

where $\tau = \max\left\{\frac{2\log(2d^2)}{C_0^2}, \exp\left(2\log\xi + \frac{2}{c}\right)\right\}$.

Remark 2 *The proof of Theorem 2 is almost identical to that of the centralized version with a notable difference. Specifically, it pertains to the communication process, whereby each agent interacts exclusively with an agent at every step. Consequently, it behooves us to assign $N = 2$ in equation (10) in Appendix B and proceed with the remaining steps in a similar manner.*

5 Experimental Results

In this section, we evaluate our methods described in Section 3 in the context of solving a sparse linear bandit problem. Our theoretical analysis, as outlined in 1 and 2, demonstrates regret of order $\mathcal{O}(s_0 \log d + s_0\sqrt{T})$ which is comparable to the state-of-the-art lasso-bandit algorithms. To evaluate our approach numerically,

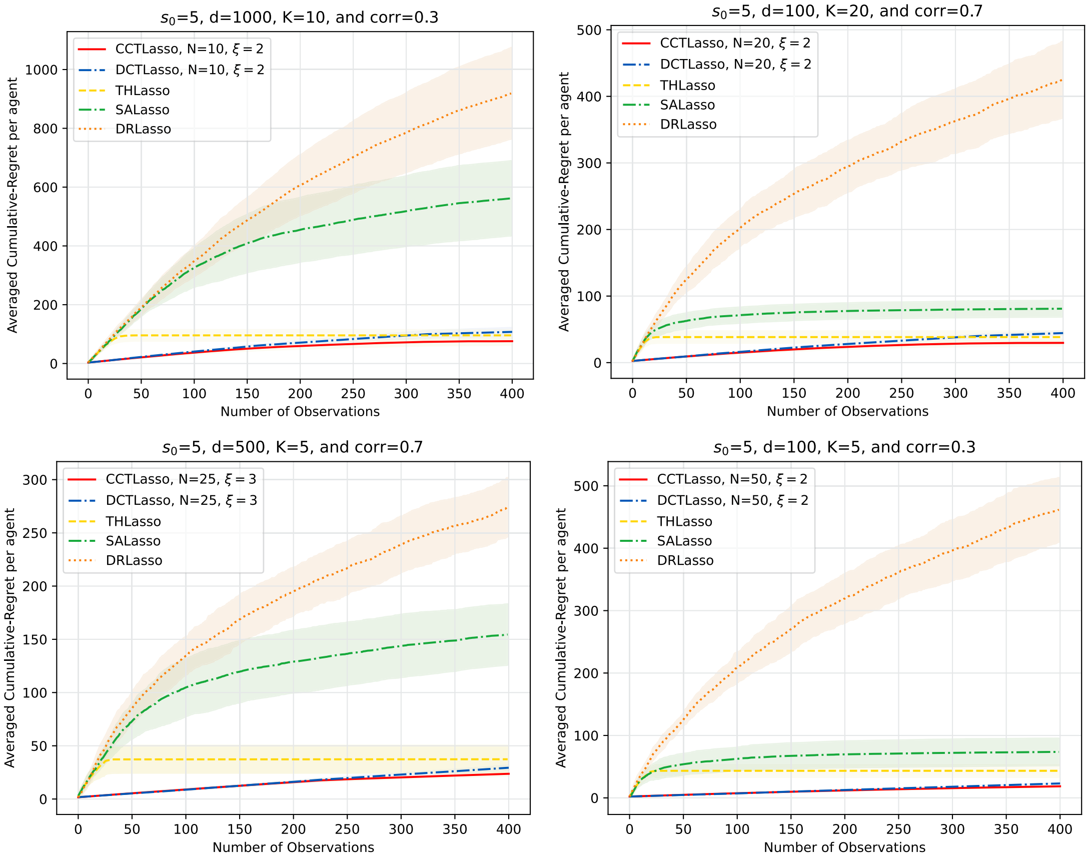

Figure 1. **Synthetic Data:** Comparison of CCTL and DCTL algorithms with state-of-the-art single-agent sparse linear bandit algorithms. The x-axis represents the number of observations per agent.

we conduct comparative experiments using both synthetic and real-world data.[1]

5.1 Synthetic Data

We focus on scenarios with $\theta^* \in \mathbb{R}^d$ is s_0-sparse. Specifically, we generate each non-zero element of θ^* in an i.i.d. fashion using a uniform distribution on the interval $[0.5, 2]$. Notably, parameter θ^* is the same for all agents. Given that, every component of the context distribution is endowed with a bounded density, Assumption 1 holds. For each round t and every agent i, we create $\mathcal{A}_t^i$ by sampling from a Gaussian distribution with mean zero and covariance matrix V. Here, for every j, $V_{j,j} = 1$ and for every $j \neq k$, $V_{j,k} = \rho^2$. We then normalize each $A_{t,k}^i$ such that its infinity-norm is at most $s_A = 5$ for all $k \in [K]$. Importantly, the feature vector components correlate over $[d]$ and $[K]$, and the Gram matrix's minimum eigenvalue is bounded below by a constant. Consequently, Assumptions 2 and 5 hold. Additionally, the symmetry of the distribution confirms Assumption 3. When the distribution is independent over arms, Proposition 1 in [24] confirms Assumption 4. It is worth noting that all agents share the $\mathcal{N}(0_K, V)$ distribution. Moreover, the additive noise is Gaussian, with i.i.d. realizations over rounds: $\omega_t^i \sim \mathcal{N}(0, 0.05)$. Furthermore, in the DCTL bandit algorithm, agents communicate through a network, which we model by a random connected graph $G = (N, E)$, by selecting the number of edges $|E|$ uniformly between $N-1$ and $2 \times N$.

5.1.1 Compare with Single-Agent Algorithms

To evaluate the effectiveness of the CCTL and DCTL bandit algorithms, we firstly compare their performance against several single-agent algorithms, including the TH Lasso bandit [4], SA Lasso bandit [24], and DR Lasso bandit [17]. We fine-tune the hyper-parameter λ_0 in the range of [0.01, 0.5] for the CCTL bandit, DCTL bandit, SA Lasso bandit, and TH Lasso bandit algorithms to optimize their performance, while for DR Lasso bandit, we utilize the hyper-parameters provided in their respective code implementations. We conduct experiments by varying the values of K, d, s_0, and ρ^2, and report the results over 10 instances for each experimental setting. The averaged cumulative regret per agent is presented in Figure 1. Our results demonstrate that DCTL and CCTL bandit algorithms outperform the other algorithms in all scenarios, with the centralized approach performing slightly better. This finding aligns with the theoretical analysis that suggest the performance of the decentralized and centralized versions of cooperative thresholded Lasso are similar.

5.1.2 Compare with Multi-Agent Algorithm

We compare our proposed method with the multi-agent low dimensional Linear Bandit method, namely, SubGoss [9]. It assumes that

[1] You can access the implementations by following the link: https://github.com/HaniyehBarghi/CooperativeThresholdedLasso.

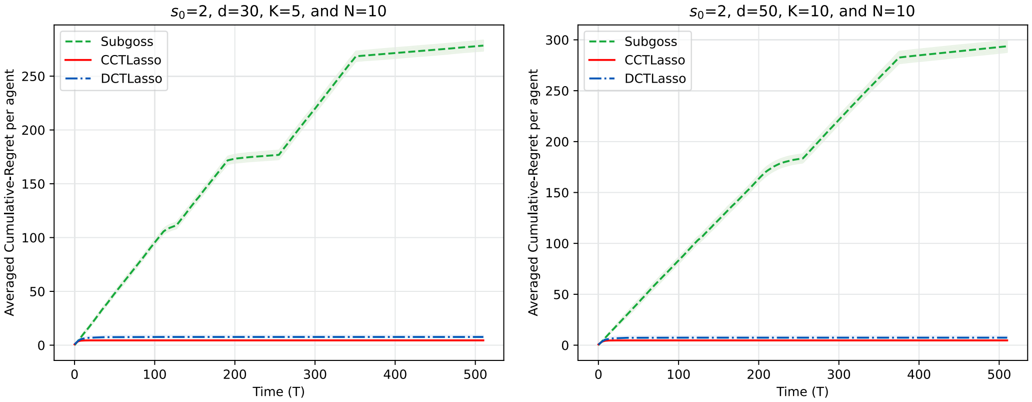

Figure 2. Comparison of CCTL and DCTL algorithms with Subgoss[9]. The x-axis shows the number of observations per agent.

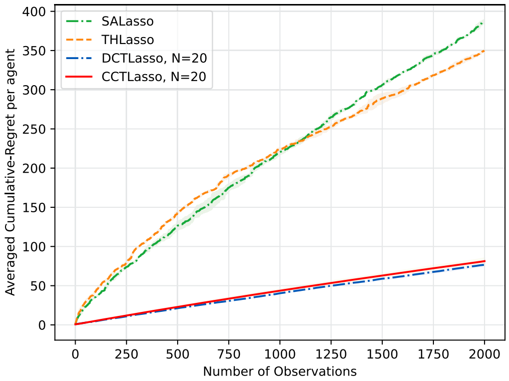

Figure 3. **Real Data:** Comparison of CCTL and DCTL algorithms with state-of-the-art single-agent sparse linear bandit algorithms. The x-axis represents the number of observations per agent.

an unknown parameter θ^* lies in one of many low-dimensional subspaces. Agents identify a small active set of subspaces and play actions only within this set, using pure exploration to identify the most likely subspace and then playing a projected version of the LinUCB algorithm to minimize regret within that subspace. The active set of subspaces is updated through collaboration and communication among agents. The algorithm has two phases in which the active subspaces remain fixed. In contrast to our problem setting, in this method, the agents have a collection of K disjoint m-dimensional subspaces, and one contains the unknown parameter θ^*. Despite the availability of such side information, our proposed method outperforms the SubGos algorithm, as demonstrated by the results of our experiments, presented in Figure 2.

5.2 Real-World Data

In this section, we demonstrate the applicability of our method on real-world datasets. We utilize Movielens 1M dataset[2], which contains approximately one million anonymous ratings from 6,000 users for 4,000 movies. We employ an SVD transformation with a dimensionality of $d = 70$. In each round, for each agent, we randomly suggest $K = 30$ movies. Agents use a bandit algorithm to select a movie (arm), aiming to choose the best one from 30 choices that satisfy the general preferences of users. Figure 3 displays the results of the SA Lasso, TH Lasso, CCTL, and DCTL bandit algorithms. It is evident that the CCTL and DCTL bandit algorithms performed well in comparison to the other approaches. As discussed in the previous section, SubGoss has several limitations and its performance is not adequate. Therefore, we do not include it in the current comparison.

6 Conclusion

In this paper, we introduce a method for solving the multi-agent sparse contextual linear bandit problem. Our approach leverages Lasso regression to reduce the problem's dimensions and utilizes the network structure to enable each agent to independently estimate the key dimensions and share this knowledge with others in only logarithmic time steps. Notably, our algorithm is the first to tackle row-wise distributed data in sparse linear bandits and delivers performance comparable to state-of-the-art single and multi-agent methods. This method has broad applicability for high-dimensional multi-agent problems, where efficient feature extraction is crucial for minimizing regret. Furthermore, we demonstrate that our proposed method achieves the same regret bound as [4] approach while only performing dimension reduction in logarithmic time steps and a single thresholding stage, as opposed to the approach proposed in [4] which performs dimension reduction in every time step. However, our theoretical analysis is limited by the way in which the threshold is defined. Our experimental results indicate that performance is not affected significantly by selecting all non-zero dimensions, whereas our theoretical approach requires a threshold to recover dimensions. Future research could explore ways to improve the theoretical framework to remove the dependency on the threshold value.

Acknowledgements

This work was supported by the German Research Foundation under Grant MA 7111/7-1 and Grant MA 7111/6-1.

References

[1] Yasin Abbasi-Yadkori, Dávid Pál, and Csaba Szepesvári, 'Improved algorithms for linear stochastic bandits', *Advances in neural information processing systems*, **24**, (2011).

[2] Data is available at https://grouplens.org/datasets/movielens/1m/.

[2] Yasin Abbasi-Yadkori, David Pal, and Csaba Szepesvari, 'Online-to-confidence-set conversions and application to sparse stochastic bandits', in *Proceedings of the Fifteenth International Conference on Artificial Intelligence and Statistics*, volume 22 of *Proceedings of Machine Learning Research*, pp. 1–9. PMLR, (21–23 Apr 2012).

[3] Sanae Amani and Christos Thrampoulidis, 'Decentralized multi-agent linear bandits with safety constraints', *Proceedings of the AAAI Conference on Artificial Intelligence*, **35**(8), 6627–6635, (May 2021).

[4] Kaito Ariu, Kenshi Abe, and Alexandre Proutière, 'Thresholded lasso bandit', in *International Conference on Machine Learning*, pp. 878–928. PMLR, (2022).

[5] Van De Geer Bühlmann, *Statistics for high-dimensional data: methods, theory and applications*, Springer Science & Business Media, 2011.

[6] Leonardo Cella and Massimiliano Pontil, 'Multi-task and meta-learning with sparse linear bandits', in *Uncertainty in Artificial Intelligence*, pp. 1692–1702. PMLR, (2021).

[7] Mithun Chakraborty, Kai Yee Phoebe Chua, Sanmay Das, and Brendan Juba, 'Coordinated versus decentralized exploration in multi-agent multi-armed bandits', in *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI-17*, pp. 164–170, (2017).

[8] Ronshee Chawla, Abishek Sankararaman, Ayalvadi Ganesh, and Sanjay Shakkottai, 'The gossiping insert-eliminate algorithm for multi-agent bandits', in *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*, volume 108 of *Proceedings of Machine Learning Research*, pp. 3471–3481. PMLR, (26–28 Aug 2020).

[9] Ronshee Chawla, Abishek Sankararaman, and Sanjay Shakkottai, 'Multi-agent low-dimensional linear bandits', *IEEE Transactions on Automatic Control*, 1–1, (2022).

[10] Rémy Degenne, Pierre M'enard, Xuedong Shang, and Michal Valko, 'Gamification of pure exploration for linear bandits', in *International Conference on Machine Learning*, (2020).

[11] Abhimanyu Dubey and Alex Pentland, 'Kernel methods for cooperative multi-agent contextual bandits', in *Proceedings of the 37th International Conference on Machine Learning*, ICML'20. JMLR.org, (2020).

[12] Abhimanyu Dubey and Alex 'Sandy' Pentland, 'Cooperative multi-agent bandits with heavy tails', in *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 2730–2739. PMLR, (13–18 Jul 2020).

[13] Abhimanyu Dubey and AlexSandy' Pentland, 'Differentially-private federated linear bandits', *Advances in Neural Information Processing Systems*, **33**, 6003–6014, (2020).

[14] Avishek Ghosh, Abishek Sankararaman, and Kannan Ramchandran. Adaptive clustering and personalization in multi-agent stochastic linear bandits, 2021.

[15] Davis Gilton and Rebecca Willett, 'Sparse linear contextual bandits via relevance vector machines', in *2017 International Conference on Sampling Theory and Applications (SampTA)*, pp. 518–522. IEEE, (2017).

[16] Yassir Jedra and Alexandre Proutiere, 'Optimal best-arm identification in linear bandits', in *Advances in Neural Information Processing Systems*, volume 33, pp. 10007–10017. Curran Associates, Inc., (2020).

[17] Gi-Soo Kim and Myunghee Cho Paik, 'Doubly-robust lasso bandit', *Advances in Neural Information Processing Systems*, **32**, (2019).

[18] Nathan Korda, Balázs Szörényi, and Shuai Li, 'Distributed clustering of linear bandits in peer to peer networks', in *Proceedings of the 33rd International Conference on International Conference on Machine Learning - Volume 48*, ICML'16, p. 1301–1309. JMLR.org, (2016).

[19] T. Lattimore and C. Szepesvári, *Bandit Algorithms*, Cambridge University Press, 2020.

[20] Lihong Li, Yu Lu, and Dengyong Zhou, 'Provably optimal algorithms for generalized linear contextual bandits', in *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, ICML'17, p. 2071–2080. JMLR.org, (2017).

[21] Shuai Li, Wei Chen, Shuai Li, and Kwong-Sak Leung, 'Improved algorithm on online clustering of bandits', in *Proceedings of the 28th International Joint Conference on Artificial Intelligence*, pp. 2923–2929, (2019).

[22] Setareh Maghsudi and Ekram Hossain, 'Multi-armed bandits with application to 5g small cells', *IEEE Wireless Communications*, **23**(3), 64–73, (2016).

[23] H. B. McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Agüera y Arcas, 'Communication-efficient learning of deep networks from decentralized data', in *International Conference on Artificial Intelligence and Statistics*, (2016).

[24] Min-hwan Oh, Garud Iyengar, and Assaf Zeevi, 'Sparsity-agnostic lasso bandit', in *International Conference on Machine Learning*, pp. 8271–8280. PMLR, (2021).

[25] Erick Schmidt, Nikolaos Gatsis, and David Akopian, 'A gps spoofing detection and classification correlator-based technique using the lasso', *IEEE Transactions on Aerospace and Electronic Systems*, **56**(6), 4224–4237, (2020).

[26] Aleksandrs Slivkins, 'Contextual bandits with similarity information', in *Proceedings of the 24th annual Conference On Learning Theory*, pp. 679–702. JMLR Workshop and Conference Proceedings, (2011).

[27] Sara van de Geer, 'On tight bounds for the lasso', *Journal of Machine Learning Research*, **19**(46), 1–48, (2018).

[28] Daniel Vial, Sanjay Shakkottai, and R. Srikant, 'Robust multi-agent multi-armed bandits', in *Proceedings of the Twenty-Second International Symposium on Theory, Algorithmic Foundations, and Protocol Design for Mobile Networks and Mobile Computing*, MobiHoc '21, p. 161–170. Association for Computing Machinery, (2021).

[29] Yuanhao Wang, Jiachen Hu, Xiaoyu Chen, and Liwei Wang, 'Distributed bandit learning: Near-optimal regret with efficient communication', *arXiv preprint arXiv:1904.06309*, (2019).

[30] Jingren Wei and Shaileshh Bojja Venkatakrishnan, 'Decvi: Adaptive video conferencing on open peer-to-peer networks', in *Proceedings of the 24th International Conference on Distributed Computing and Networking*, ICDCN '23, p. 336–341. Association for Computing Machinery, (2023).

[31] Marco Wiering, 'Multi-agent reinforcement learning for traffic light control', in *ICML*, pp. 1151–1158, (2000).

[32] Shuheng Zhou, 'Thresholded lasso for high dimensional variable selection and statistical estimation', *arXiv: Statistics Theory*, (2010).

[33] Haniyeh Barghi, Xiaotong Cheng, and Setareh Maghsudi, 'Cooperative thresholded lasso for sparse linear bandit', *arXiv preprint arXiv:2305.19161*, (2023).