Distributed Online Learning via Cooperative Contextual Bandits

Cem Tekin, *Member, IEEE*, and Mihaela van der Schaar, *Fellow, IEEE*

***Abstract*—In this paper, we propose a novel framework for decentralized, online learning by many learners. At each moment of time, an instance characterized by a certain context may arrive to each learner; based on the context, the learner can select one of its own actions (which gives a reward and provides information) or request assistance from another learner. In the latter case, the requester pays a cost and receives the reward but the provider learns the information. In our framework, learners are modeled as cooperative contextual bandits. Each learner seeks to maximize the expected reward from its arrivals, which involves trading off the reward received from its own actions, the information learned from its own actions, the reward received from the actions requested of others and the cost paid for these actions—taking into account what it has learned about the value of assistance from each other learner. We develop distributed online learning algorithms and provide analytic bounds to compare the efficiency of these with algorithms with the complete knowledge (oracle) benchmark (in which the expected reward of every action in every context is known by every learner). Our estimates show that regret—the loss incurred by the algorithm—is sublinear in time. Our theoretical framework can be used in many practical applications including Big Data mining, event detection in surveillance sensor networks and distributed online recommendation systems.**

***Index Terms*—Contextual bandits, cooperative learning, distributed learning, multi-user bandits, multi-user learning, online learning.**

I. Introduction

In this paper we propose a novel framework for online learning by multiple cooperative and decentralized learners. We assume that an instance (a data unit), characterized by a context (side) information, arrives at a learner (processor) which needs to process it either by using one of its own processing functions or by requesting another learner (processor) to process it. The learner's goal is to learn online what is the best processing function which it should use such that it maximizes its total expected reward for that instance. A data stream is an ordered sequence of instances that can be read only once or a small number of times using limited computing and storage capabilities. For example, in a stream mining application, an instance can be the data unit extracted by a sensor or camera; in a wireless communication application, an instance can be a packet that needs to be transmitted. The context can be anything that provides information about the rewards to the learners. For example, in stream mining, the context can be the type of the extracted instance; in wireless communications, the context can be the channel Signal to Noise Ratio (SNR). The processing functions in the stream mining application can be the various classification functions, while in wireless communications they can be the transmission strategies for sending the packet (Note that the selection of the processing functions by the learners can be performed based on the context and not necessarily the instance). The rewards in the stream mining can be the accuracy associated with the selected classification function, and in wireless communication they can be the resulting goodput and expended energy associated with a selected transmission strategy.

To solve such distributed online learning problems, we define a new class of multi-armed bandit solutions, which we refer to as *cooperative contextual bandits*. In the considered scenario, there is a set of cooperative learners, each equipped with a set of processing functions (arms[1]) which can be used to process the instance. By definition, cooperative learners agree to follow the rules of a prescribed algorithm provided by a designer given that the prescriped algorithm meets the set of constraints imposed by the learners. For instance, these constraints can be privacy constraints, which limits the amount of information a learner knows about the arms of the other learners. We assume a discrete time model $t = 1, 2, \ldots$, where different instances and associated context information arrive to a learner.[2] Upon the arrival of an instance, a learner needs to select either one of its arms to process the instance or it can call another learner which can select one of its own arms to process the instance and incur a cost (e.g., delay cost, communication cost, processing cost, money). Based on the selected arm, the learner receives a random reward, which is drawn from some unknown distribution that depends on the context information characterizing the instance. The goal of a learner is to maximize its total undiscounted reward up to any time horizon T. A learner does not know the expected reward (as a function of the context) of its own arms or of the other learners' arms. In fact, we go one step further and assume that a

Manuscript received August 25, 2013; revised April 09, 2014, December 12, 2014, and March 21, 2015; accepted April 16, 2015. Date of publication May 07, 2015; date of current version June 08, 2015. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Sergios Theodoridis. The work is partially supported by the grants NSF CNS 1016081 and AFOSR DDDAS. A preliminary version of this work appeared in Allerton 2013.

The authors are with the Department of Electrical Engineering, University of California, Los Angeles (UCLA), Los Angeles, CA 90095-1594 USA (e-mail: cmtkn@ucla.edu; mihaela@ee.ucla.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TSP.2015.2430837

[1]We use the terms action and arm interchangeably.

[2]Assuming synchronous agents/learners is common in the decentralized multi-armed bandit literature [1], [2]. Although our formulation is for synchronous learners, our results directly apply to the asynchronous learners, where times of instance and context arrivals can be different. A learner may not receive an instance and context at every time slot t. Then, instead of the final time T, our performance bounds for learner i will depend on the total number of arrivals to learner i by time T.

learner does not know anything about the set of arms available to other learners except an upper bound on the number of their arms. The learners are cooperative because they obtain mutual benefits from cooperation—a learner's benefit from calling another learner may be an increased reward as compared to the case when it uses solely its own arms; the benefit of the learner asked to perform the processing by another learner is that it can learn about the performance of its own arm based on its reward for the calling learner. This is especially beneficial when certain instances and associated contexts are less frequent, or when gathering labels (observing the reward) is costly.

The problem defined in this paper is a generalization of the well-known contextual bandit problem [3]–[8], in which there is a single learner who has access to all the arms. However, the considered distributed online learning problem is significantly more challenging because a learner cannot observe the arms of other learners and cannot directly estimate the expected rewards of those arms. Moreover, the heterogeneous contexts arriving at each learner lead to different learning rates for the various learners. We design distributed online learning algorithms whose long-term average rewards converge to the best distributed solution which can be obtained if we assumed complete knowledge of the expected arm rewards of each learner for each context.

To rigorously quantify the learning performance, we define the regret of an online learning algorithm for a learner as the difference between the expected total reward of the best decentralized arm selection scheme given complete knowledge about the expected arm rewards of all learners and the expected total reward of the algorithm used by the learner. Simply, the regret of a learner is the loss incurred due to the unknown system dynamics compared to the complete knowledge benchmark. We prove a sublinear upper bound on the regret, which implies that the average reward converges to the optimal average reward. The upper bound on regret gives a lower bound on the convergence rate to the optimal average reward.

The proposed framework can be used in numerous applications including the ones given below.

1) Example 1: Consider a distributed recommender system in which there is a group of agents (learners) that are connected together via a fixed network, each of whom experiences inflows of users to its page. Each time a user arrives, an agent chooses from among a set of items (arms) to offer to that user, and the user will either reject or accept each item. When choosing among the items to offer, the agent is uncertain about the user's acceptance probability of each item, but the agent is able to observe specific background information about the user (context), such as the user's gender, location, age, etc. Users with different backgrounds will have different probabilities of accepting each item, and so the agent must learn this probability over time by making different offers. In order to promote cooperation within this network, we let each agent also recommend items of other agents to its users in addition to its own items. Hence, if the agent learns that a user with a particular context is unlikely to accept any of the agent's items, it can recommend to the user items of another agent that the user might be interested in. The agent can get a commission from the other agent if it sells the item of the other agent. This provides the necessary incentive to cooperate. However, since agents are decentralized, they do not directly share the information that they learn over time about user preferences for their own items. Hence the agents must learn about other agent's acceptance probabilities through their own trial and error.

2) Example 2: Consider a network security scenario in which autonomous systems (ASs) collaborate with each other to detect cyber-attacks. Each AS has a set of security solutions which it can use to detect attacks. The contexts are the characteristics of the data traffic in each AS. These contexts can provide valuable information about the occurrence of cyber-attacks. Since the nature of the attacks are dynamic, non-stochastic and context dependent, the efficiency of the various security solutions are dynamically varying, context dependent and unknown a-priori. Based on the extracted contexts (e.g., key properties of its traffic, the originator of the traffic etc.), an AS i may route its incoming data stream (or only the context information) to another AS j, and if AS j detects a malicious activity based on its own security solutions, it warns AS i. Due to the privacy or security concerns, AS i may not know what security applications AS j is running. This problem can be modeled as a cooperative contextual bandit problem in which the various ASs cooperate with each other to learn online which actions they should take or which other ASs they should request to take actions in order to accurately detect attacks (e.g., minimize the mis-detection probability of cyber-attacks).

The remainder of the paper is organized as follows. In Section II we describe the related work and highlight the differences from our work. In Section III we describe the choices of learners, rewards, complete knowledge benchmark, and define the regret of a learning algorithm. A cooperative contextual learning algorithm that uses a non-adaptive partition of the context space is proposed and a sublinear bound on its regret is derived in Section IV. Another learning algorithm that adaptively partitions the context space of each learner is proposed in Section V, and its regret is bounded for different types of context arrivals. In Section VI we discuss the necessity of *training phase* which is a property of both algorithms and compare them. Finally, the concluding remarks are given in Section VII.

II. Related Work

Contextual bandits have been studied before in [5]–[8] in a single agent setting, where the agent sequentially chooses from a set of arms with unknown rewards, and the rewards depend on the context information provided to the agent at each time slot. The goal of the agent is to maximize its reward by balancing exploration of arms with uncertain rewards and exploitation of the arm with the highest estimated reward. The algorithms proposed in these works are shown to achieve sublinear in time regret with respect to the complete knowledge benchmark, and the sublinear regret bounds are proved to match with lower bounds on the regret up to logarithmic factors. In all the prior work, the context space is assumed to be large and a known similarity metric over the contexts is exploited by the algorithms to estimate arm rewards together for groups of similar contexts. Groups of contexts are created by partitioning the context space. For example, [7] proposed an epoch-based uniform partition of the context space, while [5] proposed a non-uniform adaptive partition. In [9], contextual bandit methods are developed for personalized news articles recommendation and a variant of the

UCB algorithm [10] is designed for linear payoffs. In [11], contextual bandit methods are developed for data mining and a perceptron based algorithm that achieves sublinear regret when the instances are chosen by an adversary is proposed. To the best of our knowledge, our work is the first to provide rigorous solutions for online learning by multiple cooperative learners when context information is present and propose a novel framework for cooperative contextual bandits to solve this problem.

Another line of work [3], [4] considers a single agent with a large set of arms (often uncountable). Given a similarity structure on the arm space, they propose online learning algorithms that adaptively partition the arm space to get sublinear regret bounds. The algorithms we design in this paper also exploits the similarity information, but in the context space rather than the action space, to create a partition and learn through the partition. However, distributed problem formulation, creation of the partitions and how learning is performed is very different from related prior work [3]–[8].

Previously, distributed multi-user learning is only considered for multi-armed bandits with finite number of arms and no context. In [1], [12] distributed online learning algorithms that converge to the optimal allocation with logarithmic regret are proposed for the i.i.d. arm reward model, given that the optimal allocation is an orthogonal allocation in which each user selects a different arm. Considering a similar model but with Markov arm rewards, logarithmic regret algorithms are proposed in [13], [14], where the regret is with respect to the best static policy which is not generally optimal for Markov rewards. This is generalized in [2] to dynamic resource sharing problems and logarithmic regret results are also proved for this case.

A multi-armed bandit approach is proposed in [15] to solve *decentralized constraint optimization problems* (DCOPs) with unknown and stochastic utility functions. The goal in this work is to maximize the total cumulative reward, where the cumulative reward is given as a sum of *local* utility functions whose values are controlled by variable assignments made (actions taken) by a subset of agents. The authors propose a message passing algorithm to efficiently compute a global upper confidence bound on the joint variable assignment, which leads to logarithmic in time regret. In contrast, in our formulation we consider a problem in which rewards are driven by contexts, and the agents do not know the set of actions of the other agents. In [16] a combinatorial multi-armed bandit problem is proposed in which the reward is a linear combination of a set of coefficients of a multi-dimensional action vector and an instance vector generated by an unknown i.i.d. process. They propose an upper confidence bound algorithm that computes a global confidence bound for the action vector which is the sum of the upper confidence bounds computed separately for each dimension. Under the proposed i.i.d. model, this algorithm achieves regret that grows logarithmically in time and polynomially in the dimension of the vector.

We provide a detailed comparison between our work and related work in multi-armed bandit learning in Table I. Our cooperative contextual learning framework can be seen as an important extension of the centralized contextual bandit framework [3]–[8]. The main differences are: (i) *training* phase which is required due to the informational asymmetries between learners, (ii) separation of exploration and exploitation over time instead of using an index for each arm to balance them, resulting in three-phase learning algorithms with *training*, *exploration* and *exploitation* phases, (iii) coordinated context space partitioning in order to balance the differences in reward estimation due to heterogeneous context arrivals to the learners. Although we consider a three-phase learning structure, our learning framework can work together with index-based policies such as the ones proposed in [5], by restricting the index updates to time slots that are not in the training phase. Our three-phase learning structure separates exploration and exploitation into distinct time slots, while they take place concurrently for an index-based policy. We will discuss the differences between these methods in Section VI. We will also show in Section VI that the training phase is necessary for the learners to form correct estimates about each other's rewards in cooperative contextual bandits.

TABLE I
COMPARISON WITH RELATED WORK IN MULTI-ARMED BANDITS

	[5]–[8]	[2], [12], [21]	This work
Multi-user	no	yes	yes
Cooperative	N/A	yes	yes
Contextual	yes	no	yes
Context arrival process	arbitrary	N/A	arbitrary
Synchronous (syn)/ Asynchronous (asn)	N/A	syn	both
Regret	sublinear	logarithmic	sublinear

Different from our work, distributed learning is also considered in online convex optimization setting [17]–[19]. In all of these works local learners choose their actions (parameter vectors) to minimize the global total loss by exchanging messages with their neighbors and performing subgradient descent. In contrast to these works in which learners share information about their actions, the learners in our model does not share any information about their own actions. The information shared in our model is the context information of the calling learner and the reward generated by the arm of the called learner. However, this information is not shared at every time slot, and the rate of information sharing between learners who cannot help each other to gain higher rewards goes to zero asymptotically.

In addition to the aforementioned prior work, in our recent work [20] we consider online learning in a decentralized social recommender system. In this related work, we address the challenges of decentralization, cooperation, incentives and privacy that arises in a network of recommender systems. We model the item recommendation strategy of a learner as a combinatorial learning problem, and prove that learning is much faster when the purchase probabilities of the items are independent of each other. In contrast, in this work we propose the general theoretical model of cooperative contextual bandits which can be applied in a variety of decentralized online learning settings including wireless sensor surveillance networks, cognitive radio networks, network security applications, recommender systems, etc. We show how context space partition can be adapted based on the context arrival process and prove the necessity of the training phase.

III. PROBLEM FORMULATION

The system model is shown in Fig. 1. There are M learners which are indexed by the set $\mathcal{M} = \{1, 2, \ldots, M\}$. Let $\mathcal{M}_{-i} := \mathcal{M} - \{i\}$ be the set of learners learner i can choose from to

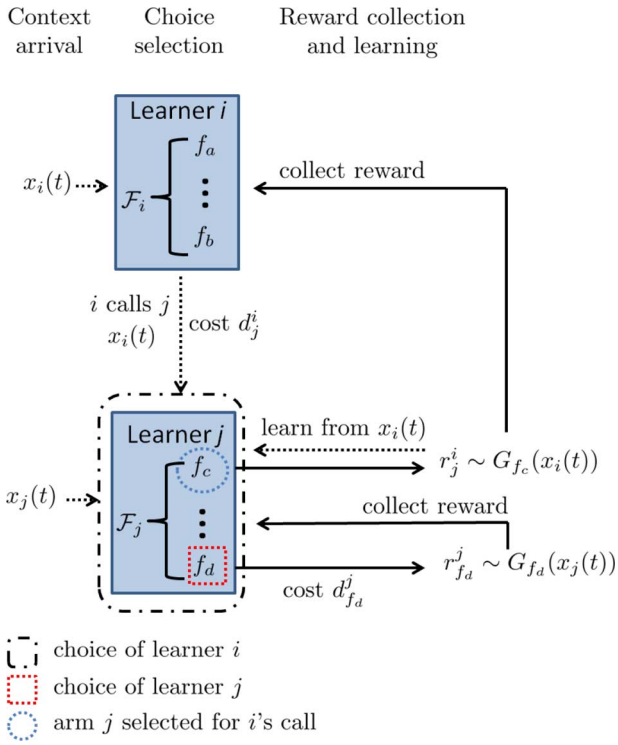

Fig. 1. System model from the viewpoint of learners i and j. Here i exploits j to obtain a high reward while helping j to learn about the reward of its own arm.

receive a reward. Let $\mathcal{F}_i$ denote the set of *arms* of learner i. Let $\mathcal{F} := \cup_{j\in\mathcal{M}}\mathcal{F}_j$ denote the *set of all arms*. Let $\mathcal{K}_i := \mathcal{F}_i \cup \mathcal{M}_{-i}$. We call $\mathcal{K}_i$ the set of *choices* for learner i. We use index k to denote any choice in $\mathcal{K}_i$, f to denote arms of the learners, j to denote other learners in $\mathcal{M}_{-i}$. Let $M_i := |\mathcal{M}_{-i}|$, $F_i := |\mathcal{F}_i|$ and $K_i := |\mathcal{K}_i|$, where $|\cdot|$ is the cardinality operator. A summary of notations is provided in Appendix B.

The learners operate under the following privacy constraint: A learner's set of arms is its private information. This is important when the learners want to cooperate to maximize their rewards, but do not want to reveal their technology/methods. For instance in stream mining, a learner may not want to reveal the types of classifiers it uses to make predictions, or in network security a learner may not want to reveal how many nodes it controls in the network and what types of security protocols it uses. However, each learner knows an upper bound on the number of arms the other learners have. Since the learners are cooperative, they can follow the rules of any learning algorithm as long as the proposed learning algorithm satisfies the privacy constraint. In this paper, we design such a learning algorithm and show that it is optimal in terms of average reward.

These learners work in a discrete time setting $t = 1, 2, \ldots, T$, where the following events happen sequentially, in each time slot: (i) an instance with context $x_i(t)$ arrives to each learner $i \in \mathcal{M}$; (ii) based on $x_i(t)$, learner i either chooses one of its arms $f \in \mathcal{F}_i$ or calls another learner and sends $x_i(t)$;[3] (iii) for each learner who called learner i at time t, learner i chooses one of its arms $f \in \mathcal{F}_i$; (iv) learner i observes the rewards of all the arms $f \in \mathcal{F}_i$ it had chosen both for its own contexts and for other learners; (v) learner i either obtains directly the reward of its own arm it had chosen, or a reward that is passed from the learner that it had called for its own context.[4]

The contexts $x_i(t)$ come from a bounded D dimensional space $\mathcal{X}$, which is taken to be $[0,1]^D$ without loss of generality. When selected, an arm $f \in \mathcal{F}$ generates a random reward sampled from an unknown, context dependent distribution $G_f(x)$ with support in $[0,1]$.[5] The expected reward of arm $f \in \mathcal{F}$ for context $x \in \mathcal{X}$ is denoted by $\pi_f(x)$. Learner i incurs a known deterministic and fixed cost d^i_k for selecting choice $k \in \mathcal{K}_i$.[6] For example for $k \in \mathcal{F}_i$, d^i_k can represent the cost of activating arm k, while for $k \in \mathcal{M}_{-i}$, d^i_k can represent the cost of communicating with learner k and/or the payment made to learner k. Although in our system model we assume that each learner i can directly call another learner j, our model can be generalized to learners over a network where calling learners that are away from learner i has a higher cost for learner i. Learner i knows the set of other learners $\mathcal{M}_{-i}$ and costs of calling them, i.e., $d^i_j, j \in \mathcal{M}_{-i}$, but does not know the set of arms $\mathcal{F}_j$, $j \in \mathcal{M}_{-i}$, but only knows an upper bound on the number of arms that each learner has, i.e., $F_{\max}$ on F_j, $j \in \mathcal{M}_{-i}$. Since the costs are bounded, without loss of generality we assume that costs are normalized, i.e., $d^i_k \in [0,1]$ for $k \in \mathcal{K}_i, i \in \mathcal{M}$. The *net reward* of learner i from a choice is equal to the obtained reward minus cost of selecting the choice. The net reward of a learner is always in $[-1,1]$.

The learners are cooperative which implies that when called by learner i, learner j will choose one of its own arms which it believes to yield the highest expected reward given the context of learner i.

The expected reward of an arm is similar for similar contexts, which is formalized in terms of a Hölder condition given in the following assumption.

Assumption 1: There exists $L > 0$, $\alpha > 0$ such that for all $f \in \mathcal{F}$ and for all $x, x' \in \mathcal{X}$, we have $|\pi_f(x) - \pi_f(x')| \leq L\|x - x'\|^\alpha$, where $\|\cdot\|$ denotes the Euclidian norm in $\mathbb{R}^D$. We assume that α is known by the learners. In the contextual bandit literature this is referred to as *similarity information* [5], [22]. Different from prior works on contextual bandit, we do not require L to be known by the learners. However, L will appear in our performance bounds.

The goal of learner i is to maximize its total expected reward. In order to do this, it needs to learn the rewards from its choices. Thus, learner i should concurrently explore the choices in $\mathcal{K}_i$ to learn their expected rewards, and exploit the best believed choice for its contexts which maximizes the reward minus cost. In the next subsection we formally define the complete knowledge benchmark. Then, we define the regret which is the performance loss due to uncertainty about arm rewards.

[3]An alternative formulation is that learner i selects multiple choices from $\mathcal{K}_i$ at each time slot, and receives sum of the rewards of the selected choices. All of the ideas/results in this paper can be extended to this case as well.

[4]Although in our problem description the learners are synchronized, our model also works for the case where instance/context arrives asynchronously to each learner.

[5]Our results can be generalized to rewards with bounded support $[b_1, b_2]$ for $-\infty < b_1 < b_2 < \infty$. This will only scale our performance bounds by a constant factor.

[6]Alternatively, we can assume that the costs are random variables with bounded support whose distribution is unknown. In this case, the learners will not learn the reward but they will learn reward minus cost which is essentially the same thing. However, our performance bounds will be scaled by a constant factor.

A. Optimal Arm Selection Policy With Complete Information

We define learner j's expected reward for context x as $\pi_j(x) := \pi_{f_j^*(x)}(x)$, where $f_j^*(x) := \arg\max_{f\in\mathcal{F}_j} \pi_f(x)$. This is the maximum expected reward learner j can provide when called by a learner with context x. For learner i, $\mu_k^i(x) := \pi_k(x) - d_k^i$ denotes the net reward of choice $k \in \mathcal{K}_i$ for context x. Our benchmark when evaluating the performance of the learning algorithms is the optimal solution which selects the choice with the highest expected net reward for learner i for its context x. This is given by

$$k_i^*(x) := \arg\max_{k\in\mathcal{K}_i} \mu_k^i(x)\ \forall x \in \mathcal{X}. \tag{1}$$

Since knowing $\mu_j^i(x)$ requires knowing $\pi_f(x)$ for $f \in \mathcal{F}_j$, knowing the optimal solution means that learner i knows the arm in $\mathcal{F}$ that yields the highest expected reward for each $x \in \mathcal{X}$.

B. The Regret of Learning

Let $a_i(t)$ be the choice selected by learner i at time t. Since learner i has no a priori information, this choice is only based on the past history of selections and reward observations of learner i. The rule that maps the history of learner i to its choices is called the learning algorithm of learner i. Let $\boldsymbol{a}(t) := (a_1(t), \ldots, a_M(t))$ be the choice vector at time t. We let $b_{i,j}(t)$ denote the arm selected by learner i when it is called by learner j at time t. If j does not call i at time t, then $b_{i,j}(t) = \emptyset$. Let $\boldsymbol{b}_i(t) = \{b_{i,j}(t)\}_{j\in\mathcal{M}_{-i}}$ and $\boldsymbol{b}(t) = \{\boldsymbol{b}_i(t)\}_{i\in\mathcal{M}}$. The regret of learner i with respect to the complete knowledge benchmark $k_i^*(x_i(t))$ given in (1) is given by

$$R_i(T) := \sum_{t=1}^{T} \left(\pi_{k_i^*(x_i(t))}(x_i(t)) - d^i_{k_i^*(x_i(t))}\right) - \mathrm{E}\left[\sum_{t=1}^{T} r^i_{a_i(t)}(x_i(t),t) - d^i_{a_i(t)}\right]$$

where $r^i_{a_i(t)}(x_i(t),t)$ denotes the random reward of choice $a_i(t) \in \mathcal{K}_i$ for context x at time t for learner i, and the expectation is taken with respect to the selections made by the distributed algorithm of the learners and the statistics of the rewards. For example, when $a_i(t) = j$ and $b_{j,i}(t) = f \in \mathcal{F}_j$, this random reward is sampled from the distribution of arm f.

Regret gives the convergence rate of the total expected reward of the learning algorithm to the value of the optimal solution given in (1). Any algorithm whose regret is sublinear, i.e., $R(T) = O(T^\gamma)$ such that $\gamma < 1$, will converge to the optimal solution in terms of the average reward. In the subsequent sections we will propose two different distributed learning algorithms with sublinear regret.

IV. A Distributed Uniform Context Partitioning Algorithm

The algorithm we consider in this section forms at the beginning a uniform partition of the context space for each learner. Each learner estimates its choice rewards based on the past history of arrivals to each set in the partition independently from the other sets in the partition. This distributed learning algorithm is called *Contextual Learning With Uniform Partition* (CLUP) and its pseudocode is given in Figs. 2–4. For learner i, CLUP is composed of two parts. The first part is the *maximization part* (see Fig. 3), which is used by learner i to maximize its reward from its own contexts. The second part is the *cooperation part* (see Fig. 4), which is used by learner i to help other learners maximize their rewards for their own contexts.

CLUP for learner i:

Input: $D_1(t)$, $D_2(t)$, $D_3(t)$, T, m_T
Initialize sets: Create partition $\mathcal{P}_T$ of $[0,1]^D$ into $(m_T)^D$ identical hypercubes
Initialize counters: $N_p^i = 0$, $\forall p \in \mathcal{P}_T$, $N^i_{k,p} = 0, \forall k \in \mathcal{K}_i, p \in \mathcal{P}_T$, $N^{\mathrm{tr},i}_{j,p} = 0, \forall j \in \mathcal{M}_{-i}, p \in \mathcal{P}_T$
Initialize estimates: $\bar{r}^i_{k,p} = 0$, $\forall k \in \mathcal{K}_i$, $p \in \mathcal{P}_T$
while $t \geq 1$ **do**
  Run CLUPmax to get choice a_i, $p = p_i(t)$ and $train$
  If $a_i \in \mathcal{M}_{-i}$ call learner a_i and pass $x_i(t)$
  Receive $\mathcal{C}_i(t)$, the set of learners who called i, and their contexts
  if $\mathcal{C}_i(t) \neq \emptyset$ **then**
    Run CLUPcoop to get arms to be selected $\boldsymbol{b}_i := \{b_{i,j}\}_{j\in\mathcal{C}_i(t)}$ and sets that the contexts lie in $\boldsymbol{p}_i := \{p_{i,j}\}_{j\in\mathcal{C}_i(t)}$
  end if
  if $a_i \in \mathcal{F}_i$ **then**
    Pay cost $d^i_{a_i}$, receive random reward r drawn from $G_{a_i}(x_i(t))$
  else
    Pay cost $d^i_{a_i}$, receive random reward r drawn from $G_{b_{a_i,i}}(x_i(t))$
  end if
  if $train = 1$ **then**
    $N^{\mathrm{tr},i}_{a_i,p}++$
  else
    $\bar{r}^i_{a_i,p} = \frac{\bar{r}^i_{a_i,p} N^i_{a_i,p} + r}{N^i_{a_i,p}+1}$
    N_p^i++, $N^i_{a_i,p}++$
  end if
  if $\mathcal{C}_i(t) \neq \emptyset$ **then**
    for $j \in \mathcal{C}_i(t)$ **do**
      Observe random reward r drawn from $G_{b_{i,j}}(x_j(t))$
      $\bar{r}^i_{b_{i,j},p_{i,j}} = \frac{\bar{r}^i_{b_{i,j},p_{i,j}} N^i_{b_{i,j},p_{i,j}} + r}{N^i_{b_{i,j},p_{i,j}}+1}$
      $N^i_{p_{i,j}}++$, $N^i_{b_{i,j},p_{i,j}}++$
    end for
  end if
  $t = t+1$
end while

Fig. 2. Pseudocode for CLUP algorithm.

Let m_T be the *slicing parameter* of CLUP that determines the number of sets in the partition of the context space $\mathcal{X}$. When m_T is small, the number of sets in the partition is small, hence the number of contexts from the past observations which can be used to form reward estimates in each set is large. However, when m_T is small, the size of each set is large, hence the variation of the expected choice rewards over each set is high. First, we will analyze the regret of CLUP for a fixed m_T and then optimize over it to balance the aforementioned tradeoff. CLUP forms a partition of $[0,1]^D$ consisting of $(m_T)^D$ sets where each set is a D-dimensional hypercube with dimensions $1/m_T \times 1/m_T \times \cdots \times 1/m_T$. We use index p to denote a set in $\mathcal{P}_T$. For learner i let $p_i(t)$ be the set in $\mathcal{P}_T$ which $x_i(t)$ belongs to.[7]

First, we will describe the maximization part of CLUP. At time slot t learner i can be in one of the three phases: *training* phase in which learner i calls another learner with its context

[7]If $x_i(t)$ is an element of the boundary of multiple sets, then it is randomly assigned to one of these sets.

CLUPmax (maximization part of CLUP) for learner i:

$train = 0$
Find the set in $\mathcal{P}_T$ that $x_i(t)$ belongs to, i.e., $p_i(t)$
Let $p = p_i(t)$
Compute the set of under-explored arms $\mathcal{F}^{\text{ue}}_{i,p}(t)$ given in (2)
if $\mathcal{F}^{\text{ue}}_{i,p}(t) \neq \emptyset$ **then**
  Select a_i randomly from $\mathcal{F}^{\text{ue}}_{i,p}(t)$
else
  Compute the set of training candidates $\mathcal{M}^{\text{ct}}_{i,p}(t)$ given in (3)
  //Update the counters of training candidates
  for $j \in \mathcal{M}^{\text{ut}}_{i,p}(t)$ **do**
    Obtain N^j_p from learner j, set $N^{\text{tr},i}_{j,p} = N^j_p - N^i_{j,p}$
  end for
  Compute the set of under-trained learners $\mathcal{M}^{\text{ut}}_{i,p}(t)$ given in (4)
  Compute the set of under-explored learners $\mathcal{M}^{\text{ue}}_{i,p}(t)$ given in (5)
  if $\mathcal{M}^{\text{ut}}_{i,p}(t) \neq \emptyset$ **then**
    Select a_i randomly from $\mathcal{M}^{\text{ut}}_{i,p}(t)$, $train = 1$
  else if $\mathcal{M}^{\text{ue}}_{i,p}(t) \neq \emptyset$ **then**
    Select a_i randomly from $\mathcal{M}^{\text{ue}}_{i,p}(t)$
  else
    Select a_i randomly from $\arg\max_{k \in \mathcal{K}_i} \bar{r}^i_{k,p} - d^i_k$
  end if
end if

Fig. 3. Pseudocode for the maximization part of CLUP algorithm.

CLUPcoop (cooperation part of CLUP) for learner i:

for $j \in \mathcal{C}_i(t)$ **do**
  Find the set in $\mathcal{P}_T$ that $x_j(t)$ belongs to, i.e., $p_{i,j}$
  Compute the set of under-explored arms $\mathcal{F}^{\text{ue}}_{i,p_{i,j}}(t)$ given in (2)
  if $\mathcal{F}^{\text{ue}}_{i,p_{i,j}}(t) \neq \emptyset$ **then**
    Select $b_{i,j}$ randomly from $\mathcal{F}^{\text{ue}}_{i,p_{i,j}}(t)$
  else
    $b_{i,j} = \arg\max_{f \in \mathcal{F}_i} \bar{r}^i_{f,p_{i,j}}$
  end if
end for

Fig. 4. Pseudocode for the cooperation part of CLUP algorithm.

such that when the reward is received, the called learner can update the estimated reward of its selected arm (but learner i does not update the estimated reward of the selected learner), *exploration* phase in which learner i selects a choice in $\mathcal{K}_i$ and updates its estimated reward, and *exploitation* phase in which learner i selects the choice with the highest estimated net reward.

Recall that the learners are cooperative. Hence, when called by another learner, learner i will choose its arm with the highest estimated reward for the calling learner's context. To gain the highest possible reward in exploitations, learner i must have an accurate estimate of other learners' expected rewards without observing the arms selected by them. In order to do this, before forming estimates about the expected reward of learner j, learner i needs to make sure that learner j will almost always select its best arm when called by learner i. Thus, the training phase of learner i helps other learners build accurate estimates about rewards of their arms, before learner i uses any rewards from these learners to form reward estimates about them. In contrast, the exploration phase of learner i helps it to build accurate estimates about rewards of its choices. These two phases indirectly help learner i to maximize its total expected reward in the long run.

Next, we define the counters learner i keeps for each set in $\mathcal{P}_T$ for each choice in $\mathcal{K}_i$, which are used to decide its current phase. Let $N^i_p(t)$ be the number of context arrivals to learner i in $p \in \mathcal{P}_T$ by time t (its own arrivals and arrivals to other learners who call learner i) except the training phases of learner i. For $f \in \mathcal{F}_i$, let $N^i_{f,p}(t)$ be the number of times arm f is selected in response to a context arriving to set p by learner i by time t (including times other learners select learner i for their contexts in set p). Other than these, learner i keeps two counters for each other learner in each set in the partition, which it uses to decide training, exploration or exploitation. The first one, i.e., $N^{\text{tr},i}_{j,p}(t)$, is an estimate on the number of context arrivals to learner j from all learners except the training phases of learner j and exploration, exploitation phases of learner i. This is an estimate because learner i updates this counter only when it needs to train learner j. The second one, i.e., $N^i_{j,p}(t)$, counts the number of context arrivals to learner j only from the contexts of learner i in set p at times learner i selected learner j in its exploration and exploitation phases by time t. Based on the values of these counters at time t, learner i either trains, explores or exploits a choice in $\mathcal{K}_i$. This three-phase learning structure is one of the major components of our learning algorithm which makes it different than the algorithms proposed for the contextual bandits in the literature which assigns an index to each choice and selects the choice with the highest index.

At each time slot t, learner i first identifies $p_i(t)$. Then, it chooses its phase at time t by giving highest priority to exploration of its own arms, second highest priority to training of other learners, third highest priority to exploration of other learners, and lowest priority to exploitation. The reason that exploration of own arms has a higher priority than training of other learners is that it can reduce the number of trainings required by other learners, which we will describe below.

First, learner i identifies its set of under-explored arms:

$$\mathcal{F}^{\text{ue}}_{i,p}(t) := \{f \in \mathcal{F}_i : N^i_{f,p}(t) \leq D_1(t)\} \tag{2}$$

where $D_1(t)$ is a deterministic, increasing function of t which is called *the control function*. We will specify this function later, when analyzing the regret of CLUP. The accuracy of reward estimates of learner i for its own arms increases with $D_1(t)$, hence it should be selected to balance the tradeoff between accuracy and the number of explorations. If this set is non-empty, learner i enters the exploration phase and randomly selects an arm in this set to explore it. Otherwise, learner i identifies the set of training candidates:

$$\mathcal{M}^{\text{ct}}_{i,p}(t) := \{j \in \mathcal{M}_{-i} : N^{\text{tr},i}_{j,p}(t) \leq D_2(t)\} \tag{3}$$

where $D_2(t)$ is a control function similar to $D_1(t)$. Accuracy of other learners' reward estimates of their own arms increase with $D_2(t)$, hence it should be selected to balance the possible reward gain of learner i due to this increase with the reward loss of learner i due to number of trainings. If this set is non-empty, learner i asks the learners $j \in \mathcal{M}^{\text{ct}}_{i,p}(t)$ to report $N^j_p(t)$. Based in the reported values it recomputes $N^{\text{tr},i}_{j,p}(t)$ as $N^{\text{tr},i}_{j,p}(t) = N^j_p(t) - N^i_{j,p}(t)$. Using the updated values, learner i identifies the set of under-trained learners:

$$\mathcal{M}^{\text{ut}}_{i,p}(t) := \{j \in \mathcal{M}_{-i} : N^{\text{tr},i}_{j,p}(t) \leq D_2(t)\}. \tag{4}$$

If this set is non-empty, learner i enters the training phase and randomly selects a learner in this set to train it.[8] When $\mathcal{M}_{i,p}^{\mathrm{ct}}(t)$ or $\mathcal{M}_{i,p}^{\mathrm{ut}}(t)$ is empty, this implies that there is no under-trained learner, hence learner i checks if there is an under-explored choice. The set of learners that are under-explored by learner i is given by

$$\mathcal{M}_{i,p}^{\mathrm{ue}}(t) := \{j \in \mathcal{M}_{-i} : N_{j,p}^{i}(t) \leq D_3(t)\} \tag{5}$$

where $D_3(t)$ is also a control function similar to $D_1(t)$. If this set is non-empty, learner i enters the exploration phase and randomly selects a choice in this set to explore it. Otherwise, learner i enters the exploitation phase in which it selects the choice with the highest estimated net reward, i.e.,

$$a_i(t) \in \arg\max_{k \in \mathcal{K}_i} \bar{r}_{k,p}^{i}(t) - d_k^i \tag{6}$$

where $\bar{r}_{k,p}^{i}(t)$ is the sample mean estimate of the rewards learner i observed (not only collected) from choice k by time t, which is computed as follows. For $j \in \mathcal{M}_{-i}$, let $\mathcal{E}_{j,p}^{i}(t)$ be the set of rewards collected by learner i at times it selected learner j while learner i's context is in set p in its exploration and exploitation phases by time t. For estimating the rewards of its own arms, learner i can also use the rewards obtained by other learners at times they called learner i. In order to take this into account, for $f \in \mathcal{F}_i$, let $\mathcal{E}_{f,p}^{i}(t)$ be the set of rewards collected by learner i at times it selected its arm f for its own contexts in set p union the set of rewards observed by learner i when it selected its arm f for other learners calling it with contexts in set p by time t. Therefore, sample mean reward of choice $k \in \mathcal{K}_i$ in set p for learner i is defined as $\bar{r}_{k,p}^{i}(t) = (\sum_{r \in \mathcal{E}_{k,p}^{i}(t)} r)/|\mathcal{E}_{k,p}^{i}(t)|$. An important observation is that computation of $\bar{r}_{k,p}^{i}(t)$ does not take into account the costs related to selecting choice k. Reward generated by an arm only depends on the context it is selected at but not on the identity of the learner for whom that arm is selected. However, the costs incurred depend on the identity of the learner. Let $\hat{\mu}_{k,p}^{i}(t) := \bar{r}_{k,p}^{i}(t) - d_k^i$ be the estimated net reward of choice k for set p. Of note, when there is more than one maximizer of (6), one of them is randomly selected. In order to run CLUP, learner i does not need to keep the sets $\mathcal{E}_{k,p}^{i}(t)$ in its memory. $\bar{r}_{k,p}^{i}(t)$ can be computed by using only $\bar{r}_{k,p}^{i}(t-1)$ and the reward at time t.

The cooperation part of CLUP operates as follows. Let $\mathcal{C}_i(t)$ be the learners who call learner i at time t. For each $j \in \mathcal{C}_i(t)$, learner i first checks if it has any under-explored arm f for $p_j(t)$, i.e., f such that $N_{f,p_j(t)}^{i}(t) \leq D_1(t)$. If so, it randomly selects one of its under-explored arms and provides its reward to learner j. Otherwise, it exploits its arm with the highest estimated reward for learner j's context, i.e.,

$$b_{i,j}(t) \in \arg\max_{f \in \mathcal{F}_i} \bar{r}_{f,p_j(t)}^{i}(t). \tag{7}$$

[8]Most of the regret bounds proposed in this paper can also be achieved by setting $N_{j,p}^{\mathrm{tr},i}(t)$ to be the number of times learner i trains learner j by time t, without considering other context observations of learner j. However, by recomputing $N_{j,p}^{\mathrm{tr},i}(t)$, learner i can avoid many unnecessary trainings especially when own context arrivals of learner j is adequate for it to form accurate estimates about its arms for set p or when learners other than learner i have already helped learner j to build accurate estimates for its arms in set p.

A. Analysis of the Regret of CLUP

Let $\beta_a := \sum_{t=1}^{\infty} 1/t^a$, and let $\log(.)$ denote logarithm in base e. For each set (hypercube) $p \in \mathcal{P}_T$ let $\overline{\pi}_{f,p} := \sup_{x \in p} \pi_f(x)$, $\underline{\pi}_{f,p} := \inf_{x \in p} \pi_f(x)$, for $f \in \mathcal{F}$, and $\overline{\mu}_{k,p}^{i} := \sup_{x \in p} \mu_k^i(x)$, $\underline{\mu}_{k,p}^{i} := \inf_{x \in p} \mu_k^i(x)$, for $k \in \mathcal{K}_i$. Let x_p^* be the context at the center (center of symmetry) of the hypercube p. We define the optimal choice of learner i for set p as $k_i^*(p) := \arg\max_{k \in \mathcal{K}_i} \mu_k^i(x_p^*)$. When the set p is clear from the context, we will simply denote the optimal choice for set p with k_i^*. Let

$$\mathcal{L}_p^i(t) := \left\{k \in \mathcal{K}_i \text{ such that } \underline{\mu}_{k_i^*(p),p}^{i} - \overline{\mu}_{k,p}^{i} > At^{\theta}\right\}$$

be the set of suboptimal choices for learner i for hypercube p at time t, where $\theta < 0$, $A > 0$ are parameters that are only used in the analysis of the regret and do not need to be known by the learners. First, we will give regret bounds that depend on values of θ and A and then we will optimize over these values to find the best bound. Also related to this let

$$\mathcal{F}_p^j(t) := \left\{f \in \mathcal{F}_j \text{ such that } \underline{\pi}_{f_j^*(p),p} - \overline{\pi}_{f,p} > At^{\theta}\right\}$$

be the set of suboptimal arms of learner j for hypercube p at time t, where $f_j^*(p) = \arg\max_{f \in \mathcal{F}_j} \pi_f(x_p^*)$. Also when the set p is clear from the context we will just use f_j^*. The arms in $\mathcal{F}_p^j(t)$ are the ones that learner j should not select when called by another learner.

The regret given in (1) can be written as a sum of three components: $R_i(T) = \mathrm{E}[R_i^e(T)] + \mathrm{E}[R_i^s(T)] + \mathrm{E}[R_i^n(T)]$, where $R_i^e(T)$ is the regret due to trainings and explorations by time T, $R_i^s(T)$ is the regret due to suboptimal choice selections in exploitations by time T and $R_i^n(T)$ is the regret due to near optimal choice selections in exploitations by time T, which are all random variables. In the following lemmas we will bound each of these terms separately. The following lemma bounds $\mathrm{E}[R_i^e(T)]$.

Lemma 1: When CLUP is run by all learners with parameters $D_1(t) = t^z \log t$, $D_2(t) = F_{\max} t^z \log t$, $D_3(t) = t^z \log t$ and $m_T = \lceil T^{\gamma} \rceil$,[9] where $0 < z < 1$ and $0 < \gamma < 1/D$, we have

$$\begin{aligned}\mathrm{E}[R_i^e(T)] &\leq \sum_{p=1}^{(m_T)^D} 2(F_i + M_i(F_{\max}+1))T^z \log T \\ &\quad + 2(F_i + 2M_i)(m_T)^D \\ &\leq 2^{D+1} Z_i T^{z+\gamma D} \log T + 2^{D+1}(F_i + 2M_i)T^{\gamma D}\end{aligned}$$

where

$$Z_i := (F_i + M_i(F_{\max}+1)). \tag{8}$$

Proof: Since time slot t is a training or an exploration slot for learner i if and only if $\mathcal{M}_{i,p_i(t)}^{\mathrm{ut}}(t) \cup \mathcal{M}_{i,p_i(t)}^{\mathrm{ue}}(t) \cup \mathcal{F}_{i,p_i(t)}^{\mathrm{ue}}(t) \neq \emptyset$, up to time T, there can be at most $\lceil T^z \log T \rceil$ exploration slots in which an arm in $f \in \mathcal{F}_i$ is selected by learner i, $\lceil F_{\max} T^z \log T \rceil$ training slots in which learner i selects learner $j \in \mathcal{M}_{-i}$, $\lceil T^z \log T \rceil$ exploration slots in which learner i selects learner $j \in \mathcal{M}_{-i}$. Since $\mu_k^i(x) = \pi_k^i(x) - d_k^i \in [-1, 1]$ for all $k \in \mathcal{K}_i$, the realized (hence expected) one slot loss

[9]For a number $r \in \mathbb{R}$, let $\lceil r \rceil$ be the smallest integer that is greater than or equal to r.

due to any choice is bounded above by 2. Hence, the result follows from summing the above terms and multiplying by 2, and the fact that $(m_T)^D \leq 2^D T^{\gamma D}$ for any $T \geq 1$. ■

From Lemma 1, we see that the regret due to explorations is linear in the number of hypercubes $(m_T)^D$, hence exponential in parameter γ and z.

For any $k \in \mathcal{K}_i$ and $p \in \mathcal{P}_T$, the sample mean $\bar{r}^i_{k,p}(t)$ represents a random variable which is the average of the independent samples in set $\mathcal{E}^i_{k,p}(t)$. Let $\Xi^i_{j,p}(t)$ be the event that a suboptimal arm $f \in \mathcal{F}_j$ is selected by learner $j \in \mathcal{M}_{-i}$, when it is called by learner i for a context in set p for the tth time in the exploitation phases of learner i. Let $X^i_{j,p}(t)$ denote the random variable which is the number of times learner j selects a suboptimal arm when called by learner i in exploitation slots of learner i when the context is in set $p \in \mathcal{P}_T$ by time t. Clearly, we have

$$X^i_{j,p}(t) = \sum_{t'=1}^{|\mathcal{E}^i_{j,p}(t)|} \mathrm{I}(\Xi^i_{j,p}(t')) \tag{9}$$

where $\mathrm{I}(\cdot)$ is the indicator function which is equal to 1 if the event inside is true and 0 otherwise. The following lemma bounds $\mathrm{E}[R^s_i(T)]$.

Lemma 2: Consider all learners running CLUP with parameters $D_1(t) = t^z \log t$, $D_2(t) = F_{\max} t^z \log t$, $D_3(t) = t^z \log t$ and $m_T = \lceil T^\gamma \rceil$, where $0 < z < 1$ and $0 < \gamma < 1/D$. For any $0 < \phi < 1$ if $t^{-z/2} + t^{\phi - z} + LD^{\alpha/2} t^{-\gamma\alpha} \leq At^\theta/2$ holds for all $t \leq T$, then we have

$$\mathrm{E}[R^s_i(T)] \leq 4(M_i + F_i)\beta_2 + 4(M_i + F_i) M_i F_{\max} \beta_2 \frac{T^{1-\phi}}{1-\phi}.$$

Proof: Consider time t. Let $\mathcal{W}^i(t) := \{\mathcal{M}^{\mathrm{ut}}_{i,p_i(t)}(t) \cup \mathcal{M}^{\mathrm{ue}}_{i,p_i(t)}(t) \cup \mathcal{F}^{\mathrm{ue}}_{i,p_i(t)}(t) = \emptyset\}$ be the event that learner i exploits at time t.

First, we will bound the probability that learner i selects a suboptimal choice in an exploitation slot. Then, using this we will bound the expected number of times a suboptimal choice is selected by learner i in exploitation slots. Note that every time a suboptimal choice is selected by learner i, since $\mu^i_k(x) = \pi^i_k(x) - d^i_k \in [-1, 1]$ for all $k \in \mathcal{K}_i$, the realized (hence expected) loss is bounded above by 2. Therefore, 2 times the expected number of times a suboptimal choice is selected in an exploitation slot bounds $\mathrm{E}[R^s_i(T)]$. Let $\mathcal{V}^i_k(t)$ be the event that choice k is chosen at time t by learner i. We have $R^s_i(T) \leq 2\sum_{t=1}^T \sum_{k \in \mathcal{L}^i_{p_i(t)}(t)} \mathrm{I}(\mathcal{V}^i_k(t), \mathcal{W}^i(t))$. Adopting the standard probabilistic notation, for two events E_1 and E_2, $\mathrm{I}(E_1, E_2)$ is equal to $\mathrm{I}(E_1 \cap E_2)$. Taking the expectation

$$\mathrm{E}[R^s_i(T)] \leq 2\sum_{t=1}^T \sum_{k \in \mathcal{L}^i_{p_i(t)}(t)} \mathrm{P}(\mathcal{V}^i_k(t), \mathcal{W}^i(t)). \tag{10}$$

Let $\mathcal{B}^i_{j,p}(t)$ be the event that at most t^ϕ samples in $\mathcal{E}^i_{j,p}(t)$ are collected from suboptimal arms of learner j in hypercube p. Let $\mathcal{B}^i(t) := \bigcap_{j \in \mathcal{M}_{-i}} \mathcal{B}^i_{j,p_i(t)}(t)$. For a set $\mathcal{A}$, let $\mathcal{A}^c$ denote the complement of that set. For any $k \in \mathcal{L}^i_{p_i(t)}(t)$, we have

$$\begin{aligned}
\{\mathcal{V}^i_k(t), \mathcal{W}^i(t)\} &\subset \left\{\hat{\mu}^i_{k,p_i(t)}(t) \geq \hat{\mu}^i_{k^*_i,p_i(t)}(t), \mathcal{W}^i(t), \mathcal{B}^i(t)\right\} \\
&\cup \left\{\hat{\mu}^i_{k,p_i(t)}(t) \geq \hat{\mu}^i_{k^*_i,p_i(t)}(t), \mathcal{W}^i(t), \mathcal{B}^i(t)^c\right\} \\
&\subset \left\{\hat{\mu}^i_{k,p_i(t)}(t) \geq \overline{\mu}^i_{k,p_i(t)} + H_t, \mathcal{W}^i(t), \mathcal{B}^i(t)\right\} \\
&\cup \left\{\hat{\mu}^i_{k^*_i,p_i(t)}(t) \leq \underline{\mu}^i_{k^*_i,p_i(t)} - H_t, \mathcal{W}^i(t), \mathcal{B}^i(t)\right\} \\
&\cup \Big\{\hat{\mu}^i_{k,p_i(t)}(t) \geq \hat{\mu}^i_{k^*_i,p_i(t)}(t), \hat{\mu}^i_{k,p_i(t)}(t) < \overline{\mu}^i_{k,p_i(t)} + H_t, \\
&\quad \hat{\mu}^i_{k^*_i,p_i(t)}(t) > \underline{\mu}^i_{k^*_i,p_i(t)} - H_t, \mathcal{W}^i(t), \mathcal{B}^i(t)\Big\} \\
&\quad \cup \{\mathcal{B}^i(t)^c, \mathcal{W}^i(t)\}
\end{aligned} \tag{11}$$

for some $H_t > 0$. This implies that

$$\begin{aligned}
&\mathrm{P}\left(\mathcal{V}^i_k(t), \mathcal{W}^i(t)\right) \\
&\leq \mathrm{P}\left(\hat{\mu}^i_{k,p_i(t)}(t) \geq \overline{\mu}^i_{k,p_i(t)} + H_t, \mathcal{W}^i(t), \mathcal{B}^i(t)\right) \\
&\quad + \mathrm{P}\left(\hat{\mu}^i_{k^*_i,p_i(t)}(t) \leq \underline{\mu}^i_{k^*_i,p_i(t)} - H_t, \mathcal{W}^i(t), \mathcal{B}^i(t)\right) \\
&\quad + \mathrm{P}\Big(\hat{\mu}^i_{k,p_i(t)}(t) \geq \hat{\mu}^i_{k^*_i,p_i(t)}(t), \hat{\mu}^i_{k,p_i(t)}(t) < \overline{\mu}^i_{k,p_i(t)} + H_t, \\
&\quad \hat{\mu}^i_{k^*_i,p_i(t)}(t) > \underline{\mu}^i_{k^*_i,p_i(t)} - H_t, \mathcal{W}^i(t), \mathcal{B}^i(t)\Big) \\
&\quad + \mathrm{P}(\mathcal{B}^i(t)^c, \mathcal{W}^i(t)).
\end{aligned}$$

Since for any $k \in \mathcal{K}$, $\bar{\mu}^i_{k,p_i(t)} = \sup_{x \in p_i(t)} \mu^i_k(x)$, we have for any suboptimal choice $k \in \mathcal{L}^i_{p_i(t)}(t)$,

$$\begin{aligned}
&\mathrm{P}\left(\hat{\mu}^i_{k,p_i(t)}(t) \geq \overline{\mu}^i_{k,p_i(t)} + H_t, \mathcal{W}^i(t), \mathcal{B}^i(t)\right) \\
&\qquad \leq \exp(-2H_t^2 t^z \log t)
\end{aligned} \tag{12}$$

by Chernoff-Hoeffding bound since on event $\mathcal{W}^i(t)$ at least $t^z \log t$ samples are taken from each choice. Similarly, we have

$$\begin{aligned}
&\mathrm{P}\left(\hat{\mu}^i_{k^*_i,p_i(t)}(t) \leq \underline{\mu}^i_{k^*_i,p_i(t)} - H_t, \mathcal{W}^i(t), \mathcal{B}^i(t)\right) \\
&\qquad \leq \exp(-2(H_t - t^{\phi - z} - LD^{\alpha/2} m_T^{-\alpha})^2 t^z \log t)
\end{aligned} \tag{13}$$

which follows from the fact that the maximum variation of expected rewards within $p_i(t)$ is at most $LD^{\alpha/2} m_T^{-\alpha}$ and on event $\mathcal{B}^i(t)$ at most t^ϕ observations from any choice comes from a suboptimal arm of the learner corresponding to that choice. For $k \in \mathcal{L}^i_{p_i(t)}(t)$, when

$$2H_t \leq At^\theta \tag{14}$$

the three inequalities given below

$$\begin{aligned}
\underline{\mu}_{k^*_i,p_i(t)} - \overline{\mu}^i_{k,p_i(t)} &> At^\theta \\
\hat{\mu}^i_{k,p_i(t)}(t) &< \overline{\mu}^i_{k,p_i(t)} + H_t \\
\hat{\mu}^i_{k^*_i,p_i(t)}(t) &> \underline{\mu}^i_{k^*_i,p_i(t)} - H_t
\end{aligned}$$

together imply that $\hat{\mu}^i_{k,p_i(t)}(t) < \hat{\mu}^i_{k^*,p_i(t)}(t)$, which implies that

$$\begin{aligned}
&\mathrm{P}\Big(\hat{\mu}^i_{k,p_i(t)}(t) \geq \hat{\mu}^i_{k^*_i,p_i(t)}(t), \hat{\mu}^i_{k,p_i(t)}(t) < \overline{\mu}^i_{k,p_i(t)} + H_t, \\
&\qquad \hat{\mu}^i_{k^*_i,p_i(t)}(t) > \underline{\mu}^i_{k^*_i,p_i(t)} - H_t, \mathcal{W}^i(t), \mathcal{B}^i(t)\Big) = 0.
\end{aligned} \tag{15}$$

Using the results of (12) and (13) and by setting

$$\begin{aligned}
H_t &= t^{-z/2} + t^{\phi - z} + LD^{\alpha/2} t^{-\gamma\alpha} \\
&\geq t^{-z/2} + t^{\phi - z} + LD^{\alpha/2} m_T^{-\alpha}
\end{aligned} \tag{16}$$

we get

$$\mathrm{P}\left(\hat{\mu}^i_{k,p_i(t)}(t) \geq \overline{\mu}^i_{k,p_i(t)} + H_t, \mathcal{W}^i(t), \mathcal{B}^i(t)\right) \leq t^{-2} \quad (17)$$

and

$$\mathrm{P}\left(\hat{\mu}^i_{k^*_i,p_i(t)}(t) \leq \underline{\mu}^i_{k^*_i,p_i(t)} - H_t, \mathcal{W}^i(t), \mathcal{B}^i(t)\right) \leq t^{-2}. \quad (18)$$

All that is left is to bound $\mathrm{P}(\mathcal{B}^i(t)^c, \mathcal{W}^i(t))$. Applying the union bound, we have

$$\mathrm{P}(\mathcal{B}^i(t)^c, \mathcal{W}^i(t)) \leq \sum_{j\in\mathcal{M}_{-i}} \mathrm{P}(\mathcal{B}^i_{j,p_i(t)}(t)^c, \mathcal{W}^i(t)).$$

We have $\{\mathcal{B}^i_{j,p_i(t)}(t)^c, \mathcal{W}^i(t)\} = \{X^i_{j,p_i(t)}(t) \geq t^\phi\}$ (Recall $X^i_{j,p_i(t)}(t)$ from (9)). Applying the Markov inequality we have $\mathrm{P}(\mathcal{B}^i_{j,p_i(t)}(t)^c, \mathcal{W}^i(t)) \leq \mathrm{E}[X^i_{j,p_i(t)}(t)]/t^\phi$. Recall that $X^i_{j,p_i(t)}(t) = \sum_{t'=1}^{|\mathcal{E}^i_{j,p_i(t)}(t)|} \mathrm{I}(\Xi^i_{j,p_i(t)}(t'))$, and

$$\begin{aligned}
&\mathrm{P}\left(\Xi^i_{j,p_i(t)}(t)\right) \leq \sum_{m\in\mathcal{F}^j_{p_i(t)}(t)} \mathrm{P}\left(\bar{r}^j_{m,p_i(t)}(t) \geq \bar{r}^j_{f^*_j,p_i(t)}(t)\right) \\
&\leq \sum_{m\in\mathcal{F}^j_{p_i(t)}(t)} \Big(\mathrm{P}\left(\bar{r}^j_{m,p_i(t)}(t) \geq \overline{\pi}_{m,p_i(t)} + H_t, \mathcal{W}^i(t)\right) \\
&\quad + \mathrm{P}\left(\bar{r}^j_{f^*_j,p_i(t)}(t) \leq \underline{\pi}_{f^*_j,p_i(t)} - H_t, \mathcal{W}^i(t)\right) \\
&\quad + \mathrm{P}\Big(\bar{r}^j_{m,p_i(t)}(t) \geq \bar{r}^j_{f^*_j,p_i(t)}(t), \bar{r}^j_{m,p_i(t)}(t) < \overline{\pi}_{m,p_i(t)} + H_t, \\
&\quad \bar{r}^j_{f^*_j,p_i(t)}(t) > \underline{\pi}_{f^*_j,p_i(t)} - H_t, \mathcal{W}^i(t)\Big)\Big).
\end{aligned}$$

When (14) holds, the last probability in the sum above is equal to zero while the first two probabilities are upper bounded by $e^{-2(H_t)^2 t^z \log t}$. This is due to the training phase of CLUP by which it is guaranteed that every learner samples each of its own arms at least $t^z \log t$ times before learner i starts forming estimates about learner j. Therefore for any $p \in \mathcal{P}_T$, we have $\mathrm{P}\left(\Xi^i_{j,p}(t)\right) \leq \sum_{m\in\mathcal{F}^j_p(t)} 2e^{-2(H_t)^2 t^z \log t} \leq 2F_j t^{-2}$ for the value of H_t given in (16). These together imply that $\mathrm{E}[X^i_{j,p}(t)] \leq \sum_{t'=1}^{\infty} \mathrm{P}(\Xi^i_{j,p}(t')) \leq 2F_j \sum_{t'=1}^{\infty} t^{-2}$. Therefore from the Markov inequality we get

$$\mathrm{P}(\mathcal{B}^i_{j,p}(t)^c, \mathcal{W}^i(t)) = \mathrm{P}(X^i_{j,p}(t) \geq t^\phi) \leq 2F_j\beta_2 t^{-\phi}$$

for any $p \in \mathcal{P}_T$ and hence,

$$\mathrm{P}(\mathcal{B}^i(t)^c, \mathcal{W}^i(t)) \leq 2M_i F_{\max}\beta_2 t^{-\phi}. \quad (19)$$

Then, using (15), (17)–(19), we have $\mathrm{P}(\mathcal{V}^i_k(t), \mathcal{W}^i(t)) \leq 2t^{-2} + 2M_i F_{\max}\beta_2 t^{-\phi}$, for any $k \in \mathcal{L}^i_{p_i(t)}(t)$. By (10), and by the result of Appendix A, we get the stated bound for $\mathrm{E}[R^s_i(T)]$. ∎

Each time learner i calls learner j, learner j selects one of its own arms in $\mathcal{F}_j$. There is a positive probability that learner j will select one of its suboptimal arms, which implies that even if learner j is near optimal for learner i, selecting learner j may not yield a near optimal outcome. We need to take this into account, in order to bound $\mathrm{E}[R^n_i(T)]$. The next lemma bounds the expected number of such happenings.

Lemma 3: Consider all learners running CLUP with parameters $D_1(t) = t^z \log t$, $D_2(t) = F_{\max} t^z \log t$, $D_3(t) = t^z \log t$ and $m_T = \lceil T^\gamma \rceil$, where $0 < z < 1$ and $0 < \gamma < 1/D$. For any $0 < \phi < 1$ if $t^{-z/2} + t^{\phi - z} + LD^{\alpha/2} t^{-\gamma\alpha} \leq At^\theta/2$ holds for all $t \leq T$, then we have

$$\mathrm{E}[X^i_{j,p}(t)] \leq 2F_{\max}\beta_2$$

for $j \in \mathcal{M}_{-i}$.

Proof: The proof is contained within the proof of the last part of Lemma 2. ∎

We will use Lemma 3 in the following lemma to bound $\mathrm{E}[R^n_i(T)]$.

Lemma 4: Consider all learners running CLUP with parameters $D_1(t) = t^z \log t$, $D_2(t) = F_{\max} t^z \log t$, $D_3(t) = t^z \log t$ and $m_T = \lceil T^\gamma \rceil$, where $0 < z < 1$ and $0 < \gamma < 1/D$. For any $0 < \phi < 1$ if $t^{-z/2} + t^{\phi - z} + LD^{\alpha/2} t^{-\gamma\alpha} \leq At^\theta/2$ holds for all $t \leq T$, then we have

$$\begin{aligned}
\mathrm{E}[R^n_i(T)] \leq \frac{2A}{1+\theta} T^{1+\theta} + 6LD^{\alpha/2} T^{1-\alpha\gamma} \\
+ 4M_i F_{\max}\beta_2 2^D T^{\gamma D}.
\end{aligned}$$

Proof: At any time t, for any $k \in \mathcal{K}_i - \mathcal{L}^i_p(t)$ and $x \in p$, we have $\mu^i_{k^*_i(x)}(x) - \mu^i_k(x) \leq At^\theta + 3LD^{\alpha/2} T^{-\alpha\gamma}$. Similarly for any $j \in \mathcal{M}$, $f \in \mathcal{F}_j - \mathcal{F}^j_p(t)$ and $x \in p$, we have $\pi_{f^*_j(x)}(x) - \pi_f(x) \leq At^\theta + 3LD^{\alpha/2} T^{-\alpha\gamma}$.

Let $p = p_i(t)$. Due to the above inequalities, if a near optimal arm in $\mathcal{F}_i \cap (\mathcal{K}_i - \mathcal{L}^i_p(t))$ is chosen by learner i at time t, the contribution to the regret is at most $At^\theta + 3LD^{\alpha/2} T^{-\alpha\gamma}$. If a near optimal learner $j \in \mathcal{M}_{-i} \cap (\mathcal{K}_i - \mathcal{L}^i_p(t))$ is called by learner i at time t, and if learner j selects one of its near optimal arms in $\mathcal{F}_j - \mathcal{F}^j_p(t)$, then the contribution to the regret is at most $2(At^\theta + 3LD^{\alpha/2} T^{-\alpha\gamma})$. Therefore, the total regret due to near optimal choices of learner i by time T is upper bounded by

$$2\sum_{t=1}^{T} (At^\theta + 3LD^{\alpha/2} T^{-\alpha\gamma}) \leq \frac{2A}{1+\theta} T^{1+\theta} + 6LD^{\alpha/2} T^{1-\alpha\gamma}$$

by using the result in Appendix A. Each time a near optimal learner in $j \in \mathcal{M}_{-i} \cap (\mathcal{K}_i - \mathcal{L}^i_p(t))$ is called in an exploitation step, there is a small probability that the arm selected by learner j is a suboptimal one. Given in Lemma 3, the expected number of times a suboptimal arm is chosen by learner j for learner i in each hypercube p is bounded by $2F_{\max}\beta_2$. For each such choice, the one-slot regret of learner i can be at most 2, and the number of such hypercubes is bounded by $2^D T^{\gamma D}$. ∎

In the next theorem we bound the regret of learner i by combining the above lemmas.

Theorem 1: Consider all learners running CLUP with parameters $D_1(t) = t^{2\alpha/(3\alpha+D)} \log t$, $D_2(t) = F_{\max} t^{2\alpha/(3\alpha+D)} \log t$, $D_3(t) = t^{2\alpha/(3\alpha+D)} \log t$ and $m_T = \lceil T^{1/(3\alpha+D)} \rceil$. Then, we have

$$\begin{aligned}
R_i(T) &\leq 4(M_i + F_i)\beta_2 \\
&+ T^{\frac{2\alpha+D}{3\alpha+D}} \left(\frac{4(LD^{\alpha/2} + 2) + 4(M_i + F_i)M_i F_{\max}\beta_2}{(2\alpha + D)/(3\alpha + D)} \right. \\
&\left. + 6LD^{\alpha/2} + 2^{D+1} Z_i \log T \right) \\
&+ T^{\frac{D}{3\alpha+D}} (2^{D+1}(F_i + 2M_i) + 2^{D+2} M_i F_{\max}\beta_2)
\end{aligned}$$

for any sequence of context arrivals $\{x_i(t)\}_{t\in 1,\dots,T}$, $i \in \mathcal{M}$. Hence, $R_i(T) = \tilde{O}\left(MF_{\max}T^{\frac{2\alpha+D}{3\alpha+D}}\right)$, for all $i \in \mathcal{M}$, where Z_i is given in (8).

Proof: The highest orders of regret that come from trainings, explorations, suboptimal and near optimal arm selections are $\tilde{O}(T^{\gamma D+z})$, $O(T^{1-\phi})$ and $O(T^{\max\{1-\alpha\gamma,1+\theta\}})$. We need to optimize them with respect to the constraint $t^{-z/2} + t^{\phi-z} + LD^{\alpha/2}t^{-\gamma\alpha} \leq At^{\theta}/2$, $t \leq T$ which is assumed in Lemmas 2 and 4. The values that minimize the regret for which this constraint holds are $z = 2\alpha/(3\alpha + D)$, $\phi = z/2$, $\theta = -z/2$, $\gamma = z/(2\alpha)$ and $A = 2LD^{\alpha/2} + 4$. Result follows from summing the bounds in Lemmas 1, 2 and 4. ∎

Remark 1: Although the parameter m_T of CLUP depends on T and hence we require T as an input to the algorithm, we can make CLUP run independently of the final time T and achieve the same regret bound by using a well known doubling trick (see, e.g., [5]). Consider phases $\tau \in \{1, 2, \dots\}$, where each phase has length 2^{τ}. We run a new instance of algorithm CLUP at the beginning of each phase with time parameter 2^{τ}. Then, the regret of this algorithm up to any time T will be $\tilde{O}\left(T^{(2\alpha+D)/(3\alpha+D)}\right)$. Although doubling trick works well in theory, CLUP can suffer from cold-start problems. The algorithm we will define in the next section will not require T as an input parameter.

The regret bound proved in Theorem 1 is sublinear in time which guarantees convergence in terms of the average reward, i.e., $\lim_{T\to\infty} \mathrm{E}[R_i(T)]/T = 0$. For a fixed α, the regret becomes linear in the limit as D goes to infinity. On the contrary, when D is fixed, the regret decreases, and in the limit, as α goes to infinity, it becomes $O(T^{2/3})$. This is intuitive since increasing D means that the dimension of the context increases and therefore the number of hypercubes to explore increases. While increasing α means that the level of similarity between any two pairs of contexts increases, i.e., knowing the expected reward of arm f in one context yields more information about its accuracy in another context.

B. Computational Complexity of CLUP

For each set $p \in \mathcal{P}_T$, learner i keeps the sample mean of rewards from $F_i + M_i$ choices, while for a centralized bandit algorithm, the sample mean of the rewards of $|\cup_{j\in\mathcal{M}} \mathcal{F}_j|$ arms needs to be kept in memory. Since the number of sets in $\mathcal{P}_T$ is upper bounded by $2^D T^{D/(3\alpha+D)}$, the memory requirement is upper bounded by $(F_i + M_i)2^D T^{D/(3\alpha+D)}$. This means that the memory requirement is sublinearly increasing in T and thus, in the limit $T \to \infty$, required memory goes to infinity. However, CLUP can be modified so that the available memory provides an upper bound on m_T. However, in this case the regret bound given in Theorem 1 may not hold. Also the actual number of hypercubes with at least one context arrival depends on the context arrival process, hence can be very small compared to the worst-case scenario. In that case, it is enough to keep the reward estimates for these hypercubes. The following example illustrates that for a practically reasonable time frame, the memory requirement is not very high for a learner compared to a non-contextual centralized implementation (that uses partition $\{\mathcal{X}\}$). For example for $\alpha = 1$, $D = 1$, we have $2^D T^{D/(3\alpha+D)} = 2T^{1/4}$. If learner i learned through $T = 10^8$ samples, and if $M = 100$, $F_j = 100$, for all $j \in \mathcal{M}$, learner i using CLUP only needs to store at most 40000 sample mean estimates, while a standard bandit algorithm which does not exploit any context information requires to keep 10000 sample mean estimates. Although, the memory requirement is 4 times higher than the memory requirement of a standard bandit algorithm, CLUP is suitable for a distributed implementation, learner i does not require any knowledge about the arms of other learners (except an upper bound on the number of arms), and it is shown to converge to the best distributed solution.

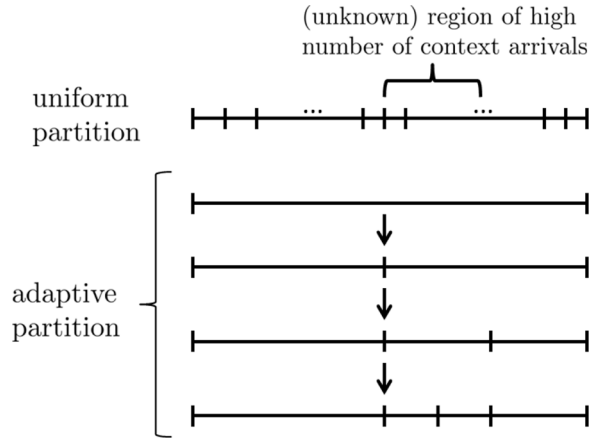

Fig. 5. An illustration showing how the partition of DCZA differs from the partition of CLUP for $D = 1$. As contexts arrive, DCZA *zooms* into regions of high number of context arrivals.

V. A Distributed Adaptive Context Partitioning Algorithm

Intuitively, the loss due to selecting a suboptimal choice for a context can be further minimized if the learners inspect the regions of $\mathcal{X}$ with large number of context arrivals more carefully, instead of using a uniform partition of $\mathcal{X}$. We do this by introducing the *Distributed Context Zooming Algorithm* (DCZA).

A. The DCZA Algorithm

In the previous section, the partition $\mathcal{P}_T$ is formed by CLUP at the beginning by choosing the slicing parameter m_T. Differently, DCZA adaptively generates the partition based on how contexts arrive. Similar to CLUP, using DCZA a learner forms reward estimates for each set in its partition based only on the history related to that set. Let $\mathcal{P}_i(t)$ be learner i's partition of $\mathcal{X}$ at time t and $p_i(t)$ denote the set in $\mathcal{P}_i(t)$ that contains $x_i(t)$. Using DCZA, learner i starts with $\mathcal{P}_i(1) = \{\mathcal{X}\}$, then divides $\mathcal{X}$ into sets with smaller sizes as time goes on and more contexts arrive. Hence the cardinality of $\mathcal{P}_i(t)$ increases with t. This division is done in a systematic way to ensure that the tradeoff between the variation of expected choice rewards inside each set and the number of past observations that are used in reward estimation for each set is balanced. As a result, the regions of the context space with a lot of context arrivals are covered with sets of smaller sizes than regions of contexts space with few context arrivals. In other words, DCZA *zooms* into the regions of context space with large number of arrivals. An illustration that shows partition of CLUP and DCZA is given in Fig. 5 for $D = 1$. As we discussed in the Section II the zooming idea have been used in a variety of multi-armed bandit problems [3]–[8], but there are differences in the problem structure and how zooming is done.

The sets in the adaptive partition of each learner are chosen from hypercubes with edge lengths coming from the set $\{1, 2^{-1}, 2^{-2}, \ldots\}$.[10] We call a D-dimensional hypercube which has edges of length 2^{-l} a level l hypercube (or level l set). For a hypercube p, let $l(p)$ denote its level. Different from CLUP, the partition of each learner in DCZA can be different since context arrivals to learners can be different. In order to help each other, learners should know about each other's partition. For this, whenever a new set of hypercubes is activated by learner i, learner i communicates this by sending the center and edge length of one of the hypercubes in the new set of hypercubes to other learners. Based on this information, other learners update their partition of learner i. Thus, at any time slot t all learners know $\mathcal{P}_i(t)$. This does not require a learner to keep M different partitions. It is enough for each learner to keep $\mathcal{P}(t) := \bigcup_{i\in\mathcal{M}} \mathcal{P}_i(t)$, which is the set of hypercubes that are active for at least one learner at time t. For $p \in \mathcal{P}(t)$ let $\tau(p)$ be the first time p is activated by one of the learners and for $p \in \mathcal{P}_i(t)$, let $\tau_i(p)$ be the first time p is activated for learner i's partition. We will describe the activation process later, after defining the counters of DCZA which are initialized and updated differently than CLUP.

$N_p^i(t)$, $p \in \mathcal{P}_i(t)$ counts the number of context arrivals to set p of learner i (from its own contexts) from times $\{\tau_i(p), \ldots, t-1\}$. For $f \in \mathcal{F}_i$, $N_{f,p}^i(t)$ counts the number of times arm f is selected in response to contexts arriving to set $p \in \mathcal{P}(t)$ (from learner i's own contexts or contexts of calling learners) from times $\{\tau(p), \ldots, t-1\}$. Similarly $N_{j,p}^{\mathrm{tr},i}(t)$, $p \in \mathcal{P}_i(t)$ is an estimate on the context arrivals to learner j in set p from all learners except the training phases of learner j and exploration, exploitation phases of learner i from times $\{\tau(p), \ldots, t-1\}$. Finally, $N_{j,p}^i(t)$ counts the number of context arrivals to learner j from exploration and exploitation phases of learner i from times $\{\tau_i(p), \ldots, t-1\}$. Let $\mathcal{E}_{f,p}^i(t)$, $f \in \mathcal{F}_i$ be the set of rewards (received or observed) by learner i at times that contribute to the increase of counter $N_{f,p}^i(t)$ and $\mathcal{E}_{j,p}^i(t)$, $j \in \mathcal{M}_{-i}$ be the set of rewards received by learner i at times that contribute to the increase of counter $N_{j,p}^i(t)$. We have $\bar{r}_{k,p}^i(t) = (\sum_{r\in\mathcal{E}_{k,p}^i(t)} r)/|\mathcal{E}_{k,p}^i(t)|$ for $k \in \mathcal{K}_i$. Training, exploration and exploitation within a hypercube p is controlled by control functions $D_1(p,t) = D_3(p,t) = 2^{2\alpha l(p)} \log t$ and $D_2(p,t) = F_{\max} 2^{2\alpha l(p)} \log t$, which depend on the level of hypercube p unlike the control functions $D_1(t)$, $D_2(t)$ and $D_3(t)$ of CLUP, which only depend on the current time. DCZA separates training, exploration and exploitation the same way as CLUP but using control functions $D_1(p,t)$, $D_2(p,t)$, $D_3(p,t)$ instead of $D_1(t)$, $D_2(t)$, $D_3(t)$.

Learner i updates its partition $\mathcal{P}_i(t)$ as follows. At the end of each time slot t, learner i checks if $N_{p_i(t)}^i(t+1)$ exceeds a threshold $2^{\rho l(p_i(t))}$, where ρ is the parameter of DCZA that is common to all learners. If $N_{p_i(t)}^i(t+1) \geq 2^{\rho l(p_i(t))}$, learner i will divide $p_i(t)$ into 2^D level $l(p_i(t))+1$ hypercubes and will note the other learners about its new partition $\mathcal{P}_i(t+1)$. With this division $p_i(t)$ is de-activated for learner i's partition. For a set p, let $\tau_i^{\mathrm{fin}}(p)$ be the time it is de-activated for learner i's partition.

[10]Hypercubes have advantages in cooperative contextual bandits because they are disjoint and a learner can pass information to another learner about its partition by only passing the center and edge length of its hypercubes.

Similar to CLUP, DCZA also have maximization and cooperation parts. The maximization part of DCZA is the same as CLUP with training, exploration and exploitation phases. The only differences are that which phase to enter is determined by comparing the counters defined above with the control functions and in exploitation phase the best choice is selected based on the sample mean estimates defined above. In the cooperation part at time t, learner i explores one of its under-explored arms or chooses its best arm for $p_j(t)$ for learner $j \in \mathcal{C}_i(t)$ using the counters and sample mean estimates defined above. Since the operation of DCZA is the same as CLUP except the differences mentioned in this section, we omitted its pseudocode to avoid repetition.

B. Analysis of the Regret of DCZA

Our analysis for CLUP in Section IV was for worst-case context arrivals. This means that the bound in Theorem 1 holds even when other learners never call learner i to train it, or other learners never learn by themselves. In this section we analyze the regret of DCZA under different types of context arrivals. Let $K_{i,l}(T)$ be the number of level l hypercubes of learner i that are activated by time T. In the following we define two extreme cases of correlation between the contexts arriving to different learners.

Definition 1: We call the context arrival process, *solo arrivals* if contexts only arrive to learner i, *identical arrivals* if $x_i(t) = x_j(t)$ for all $i, j \in \mathcal{M}$, $t = 1, \ldots, T$.

We start with a simple lemma which gives an upper bound on the highest level hypercube that is active at any time t.

Lemma 5: All the active hypercubes $p \in \mathcal{P}(t)$ at time t have at most a level of $\rho^{-1} \log_2 t + 1$.

Proof: Let $l'+1$ be the level of the highest level active hypercube. We must have $\sum_{l=0}^{l'} 2^{\rho l} < t$, otherwise the highest level active hypercube's level will be less than $l'+1$. We have, $(2^{\rho(l'+1)} - 1)/(2^\rho - 1) < t \Rightarrow 2^{\rho l'} < t \Rightarrow l' < \rho^{-1} \log_2 t$. ■

In order to analyze the regret of DCZA, we first bound the regret due to trainings and explorations in a level l hypercube. We do this for the solo and identical context arrival cases separately.

Lemma 6: Consider all learners that run DCZA with parameters $D_1(p,t) = D_3(p,t) = 2^{2\alpha l(p)} \log t$ and $D_2(p,t) = F_{\max} 2^{2\alpha l(p)} \log t$. Then, for any level l hypercube the regret of learner i due to trainings and explorations by time T is bounded above by (i) $2Z_i(2^{2\alpha l} \log T + 1)$ for solo context arrivals, (ii) $2K_i(2^{2\alpha l} \log T + 1)$ for identical context arrivals (given $F_i \geq F_j$, $j \in \mathcal{M}_{-i}$).[11]

Proof: The proof is similar to Lemma 1. Note that when the context arriving to each learner is the same and $|\mathcal{F}_i| \geq |\mathcal{F}_j|$, $j \in \mathcal{M}_{-i}$, we have $N_{j,p}^{i,tr}(t) > D_2(p,t)$ for all $j \in \mathcal{M}_{-i}$ whenever $N_{f,p}^i(t) > D_1(p,t)$ for all $f \in \mathcal{F}_i$. ■

We define the set of suboptimal choices and arms for learner i in DCZA a little differently than CLUP (suboptimality depends on the level of the hypercube but not on time), using the same notation as in the analysis of CLUP. Let

$$\mathcal{L}_p^i := \left\{ k \in \mathcal{K}_i : \underline{\mu}_{k_i^*(p),p}^i - \overline{\mu}_{k,p}^i > A^* L D^{\alpha/2} 2^{-l(p)\alpha} \right\} \quad (20)$$

[11]In order for the bound for identical context arrivals to hold for learner i we require that $F_i \geq F_j$, $j \in \mathcal{M}_{-i}$. Hence, in order for the bound for identical context arrivals to hold for all learners, we require $F_i = F_j$ for all $i, j \in \mathcal{M}$.

be the set of suboptimal choices of learner i for a hypercube p, and

$$\mathcal{F}_p^j := \left\{ f \in \mathcal{F}_j : \underline{\pi}_{f_j^*(p),p} - \overline{\pi}_{f,p} > A^* L D^{\alpha/2} 2^{-l(p)\alpha} \right\} \quad (21)$$

be the set of suboptimal arms of learner j for hypercube p, where $A^* = 2 + 4/(LD^{\alpha/2})$.

In the next lemma we bound the regret due to choosing suboptimal choices in the exploitation steps of learner i.

Lemma 7: Consider all learners running DCZA with parameters $\rho > 0$, $D_1(p,t) = D_3(p,t) = 2^{2\alpha l(p)} \log t$ and $D_2(p,t) = F_{\max} 2^{2\alpha l(p)} \log t$. Then, we have

$$\mathrm{E}[R_i^s(T)] \leq 4(M_i + F_i)\beta_2 + 4(M_i + F_i) M_i F_{\max} \beta_2 \sum_{t=1}^{T} 2^{-\alpha l(p_i(t))}.$$

Proof: The proof of this lemma is similar to the proof of Lemma 7, thus some steps are omitted. $\mathcal{W}^i(t)$ and $\mathcal{V}_k^i(t)$ are defined the same way as in Lemma 7. $\mathcal{B}_{j,p_i(t)}^i(t)$ denotes the event that at most $2^{\alpha l(p_i(t))}$ samples in $\mathcal{E}_{j,p_i(t)}^i(t)$ are collected from the suboptimal arms of learner j in $\mathcal{F}_{p_i(t)}^j$, and $\mathcal{B}^i(t) := \bigcap_{j \in \mathcal{M}_{-i}} \mathcal{B}_{j,p_i(t)}^i(t)$. We have $\mathrm{E}[R_i^s(T)] \leq 2 \sum_{t=1}^{T} \sum_{k \in \mathcal{L}_{p_i(t)}^i} \mathrm{P}(\mathcal{V}_k^i(t), \mathcal{W}^i(t))$.

Similar to Lemma 7, we have

$$\begin{aligned}
&\mathrm{P}\left(\mathcal{V}_k^i(t), \mathcal{W}^i(t)\right) \\
&\leq \mathrm{P}\left(\hat{\mu}_{k,p_i(t)}^i(t) \geq \overline{\mu}_{k,p_i(t)}^i + H_t, \mathcal{W}^i(t), \mathcal{B}^i(t)\right) \\
&\quad + \mathrm{P}\left(\hat{\mu}_{k_i^*,p_i(t)}^i(t) \leq \underline{\mu}_{k_i^*,p_i(t)}^i - H_t, \mathcal{W}^i(t), \mathcal{B}^i(t)\right) \\
&\quad + \mathrm{P}\Bigg(\hat{\mu}_{k,p_i(t)}^i(t) \geq \hat{\mu}_{k_i^*,p_i(t)}^i(t), \hat{\mu}_{k,p_i(t)}^i(t) < \overline{\mu}_{k,p_i(t)}^i + H_t, \\
&\quad \hat{\mu}_{k_i^*,p_i(t)}^i(t) > \underline{\mu}_{k_i^*,p_i(t)}^i - H_t, \mathcal{W}^i(t), \mathcal{B}^i(t)\Bigg) \\
&\quad + \mathrm{P}(\mathcal{B}^i(t)^c, \mathcal{W}^i(t)).
\end{aligned}$$

Letting

$$H_t = (LD^{\alpha/2} + 2) 2^{-\alpha l(p_i(t))}$$

we have

$$\begin{aligned}
\mathrm{P}\left(\hat{\mu}_{k,p_i(t)}^i(t) \geq \overline{\mu}_{k,p_i(t)}^i + H_t, \mathcal{W}^i(t), \mathcal{B}^i(t)\right) &\leq t^{-2} \\
\mathrm{P}\left(\hat{\mu}_{k_i^*,p_i(t)}^i(t) \leq \underline{\mu}_{k_i^*,p_i(t)}^i - H_t, \mathcal{W}^i(t), \mathcal{B}^i(t)\right) &\leq t^{-2}.
\end{aligned}$$

Since $2H_t \leq A^* L D^{\alpha/2} 2^{-l(p_i(t))\alpha}$,

$$\mathrm{P}\Bigg(\hat{\mu}_{k,p_i(t)}^i(t) \geq \hat{\mu}_{k_i^*,p_i(t)}^i(t), \hat{\mu}_{k,p_i(t)}^i(t) < \overline{\mu}_{k,p_i(t)}^i + H_t, \hat{\mu}_{k_i^*,p_i(t)}^i(t) > \underline{\mu}_{k_i^*,p_i(t)}^i - H_t, \mathcal{W}^i(t), \mathcal{B}^i(t)\Bigg) = 0.$$

Similar to the proof of Lemma 7, we have

$$\begin{aligned}
\mathrm{P}(\Xi_{j,p_i(t)}^i) &\leq 2F_j t^{-2} \\
\mathrm{E}[X_{j,p_i(t)}^i] &\leq 2F_j \beta_2 \\
\mathrm{P}(\mathcal{B}_{j,p_i(t)}^i(t)^c, \mathcal{W}^i(t)) &\leq 2F_j \beta_2 2^{-\alpha l(p_i(t))} \\
\mathrm{P}(\mathcal{B}^i(t)^c, \mathcal{W}^i(t)) &\leq 2M_i F_{\max} \beta_2 2^{-\alpha l(p_i(t))}.
\end{aligned}$$

Hence,

$$\mathrm{P}\left(\mathcal{V}_k^i(t), \mathcal{W}^i(t)\right) \leq 2t^{-2} + 2M_i F_{\max} \beta_2 2^{-\alpha l(p_i(t))}.$$

■

In the next lemma we bound the regret of learner i due to selecting near optimal choices.

Lemma 8: Consider all learners running DCZA with parameters $\rho > 0$, $D_1(p,t) = D_3(p,t) = 2^{2\alpha l(p)} \log t$ and $D_2(p,t) = F_{\max} 2^{2\alpha l(p)} \log t$. Then, we have

$$\mathrm{E}[R_i^n(T)] \leq 4M_i F_{\max} \beta_2 + 2(3 + A^*) L D^{\alpha/2} \sum_{t=1}^{T} 2^{-\alpha l(p_i(t))}.$$

Proof: For any $k \in \mathcal{K}_i - \mathcal{L}_{p_i(t)}^i$ and $x \in p_i(t)$, we have $\mu_{k_i^*(x)}^i(x) - \mu_k^i(x) \leq (3 + A^*) L D^{\alpha/2} 2^{-l(p_i(t))\alpha}$. Similarly for any $j \in \mathcal{M}$, $f \in \mathcal{F}_j - \mathcal{F}_{p_i(t)}^j(t)$ and $x \in p_i(t)$, we have $\pi_{f_j^*(x)}(x) - \pi_f(x) \leq (3 + A^*) L D^{\alpha/2} 2^{-l(p_i(t))\alpha}$.

As in the proof of Lemma 7, we have $\mathrm{P}(\Xi_{j,p_i(t)}^i(t)) \leq 2F_{\max} t^{-2}$. Thus, when a near optimal learner $j \in \mathcal{M}_{-i} \cap (\mathcal{K}_i - \mathcal{L}_p^i)$ is called by learner i at time t, the contribution to the regret from suboptimal arms of j is bounded by $4F_{\max} t^{-2}$. The one-slot regret of any near optimal arm of any near optimal learner $j \in \mathcal{M}_{-i} \cap (\mathcal{K}_i - \mathcal{L}_p^i)$ is bounded by $2(3 + A^*) L D^{\alpha/2} 2^{-l(p)\alpha}$. The one-step regret of any near optimal arm $f \in \mathcal{F}_i \cap (\mathcal{K}_i - \mathcal{L}_p^i)$ is bounded by $(3 + A^*) L D^{\alpha/2} 2^{-l(p)\alpha}$. The result is obtained by taking the sum up to time T. ■

Next, we combine the results from Lemmas 6, 7 and 8 to obtain regret bounds as a function of the number of hypercubes of each level that are activated up to time T.

Theorem 2: Consider all learners running DCZA with parameters $\rho > 0$, $D_1(p,t) = D_3(p,t) = 2^{2\alpha l(p)} \log t$ and $D_2(p,t) = F_{\max} 2^{2\alpha l(p)} \log t$. Then, for *solo arrivals*, we have

$$\begin{aligned}
R_i(T) \leq\ & 2C_1 \sum_{l=0}^{(\log_2 T/\rho)+1} K_{i,l}(T) 2^{2\alpha l} \log T \\
&+ C_2 \sum_{l=0}^{(\log_2 T/\rho)+1} K_{i,l}(T) 2^{(\rho-\alpha)l} \\
&+ 2C_1 \sum_{l=0}^{(\log_2 T/\rho)+1} K_{i,l}(T) + C_0
\end{aligned}$$

where $C_0 = 4\beta_2(M_i + F_i + M_i F_{\max})$, $C_1 = Z_i$ for *solo arrivals* and $C_1 = K_i$ for *identical arrivals* and $C_2 = 4(M_i + F_i) M_i F_{\max} \beta_2 + 2(3 + A^*) L D^{\alpha/2}$.

Proof: The result follows from summing the results of Lemmas 6, 7 and 8 and using Lemma 5. ■

Although the result in Theorem 2 bounds the regret of DCZA for an arbitrary context arrival process in terms of $K_{i,l}(T)$'s, it is possible to obtain context arrival process independent regret bounds by considering the worst-case context arrivals. The next corollary shows that the worst-case regret bound of DCZA matches with the worst-case regret bound of CLUP derived in Theorem 1.

Corollary 1: Consider all learners running DCZA with parameters $\rho = 3\alpha$, $D_1(p,t) = D_3(p,t) = 2^{2\alpha l(p)} \log t$ and

$D_2(p,t) = F_{\max} 2^{2\alpha l(p)} \log t$. Then, the worst-case regret of learner i is bounded by

$$R_i(T) \leq 2^{2(D+2\alpha)}(2C_1 \log T + C_2) T^{\frac{2\alpha+D}{3\alpha+D}} + 2C_1 2^{2D} T^{\frac{D}{3\alpha+D}} + C_0$$

where C_0, C_1 and C_2 are given in Theorem 2.

Proof: Since hypercube p remains active for at most $2^{\rho l(p)}$ context arrivals within that hypercube, combining the results of Lemmas 7 and 8, the expected loss in hypercube p in exploitation slots is at most $C_2 2^{(\rho-\alpha)l(p)}$, where C_2 is defined in Theorem 2. However, the expected loss in hypercube p due to trainings and explorations is at least $C 2^{2\alpha l(p)}$ for some constant $C > 0$, and is at most $2Z_i(2^{2\alpha l(p)} \log T + 1)$ as given in Lemma 6. In order to balance the regret due to trainings and explorations with the regret incurred in exploitation within p we set $\rho = 3\alpha$.

In the worst-case context arrivals, contexts arrive in a way that all level l hypercubes are divided into level $l+1$ hypercubes before contexts start arriving to any of the level $l+1$ hypercubes. In this way, the number of hypercubes to train and explore is maximized. Let $l_{\max}$ be the hypercube with the maximum level that had at least one context arrival on or before T in the worst-case context arrivals. We must have

$$\sum_{l=0}^{l_{\max}-1} 2^{Dl} 2^{3\alpha l} < T.$$

Otherwise, no hypercube with level $l_{\max}$ will have a context arrival by time T. From the above equation we get $l_{\max} < 1 + (\log_2 T)/(D+3\alpha)$. Thus,

$$R_i(T) \leq 2C_1 \sum_{l=0}^{l_{\max}} 2^{Dl} 2^{2\alpha l} \log T + C_2 \sum_{l=0}^{l_{\max}} 2^{Dl} 2^{2\alpha l} + 2C_1 \sum_{l=0}^{l_{\max}} 2^{Dl} + C_0.$$

■

VI. Discussion

A. Necessity of the Training Phase

In this subsection, we prove that the training phase is necessary to achieve sublinear regret for the cooperative contextual bandit problem for algorithms of the type CLUP and DCZA (without the training phase) which use (i) exploration control functions of the form $Ct^z \log t$, for constants $C > 0$, $z > 0$; (ii) form a finite partition of the context space; and (iii) use the sample mean estimator within each hypercube in the partition. We call this class of algorithms *Simple Separation of Exploration and Exploitation* (SSEE) algorithms. In order to show this, we consider a special case of expected arm rewards and context arrivals and show that independent of the rate of explorations, the regret of an SSEE algorithm is linear in time for any exploration control function $D_i(t)$[12] of the form $Ct^z \log t$ for learner i (exploration functions of learners can be different). Although, our proof does not consider index-based learning algorithms, we think that similar to our construction in Theorem 3, problem instances which will give linear regret can be constructed for any type of index policy without the training phase.

Theorem 3: Without the training phase, the regret of any SSEE algorithm is linear in time.

Proof: We will construct a problem instance for which the statement of the theorem is valid. Assume that all costs d_k^i, $k \in \mathcal{K}_i$, $i \in \mathcal{M}$ are zero. Let $M = 2$. Consider a hypercube p. We assume that at all time slots context $x^* \in p$ arrives to learner 1, and all the contexts that are arriving to learner 2 are outside p. Learner 1 has only a single arm m, learner 2 has two arms b and g. With an abuse of notation, we denote the expected reward of an arm $f \in \{m, b, g\}$ at context x^* as π_f. Assume that the arm rewards are drawn from $\{0, 1\}$ and the following is true for expected arm rewards:

$$\pi_b + C_K \delta < \pi_m < \pi_g - \delta < \pi_m + \delta \tag{22}$$

for some $\delta > 0$, $C_K > 0$, where the value of C_K will be specified later. Assume that learner 1's exploration control function is $D_1(t) = t^z \log t$, and learner 2's exploration control function is $D_2(t) = t^z \log t / K$ for some $K \geq 1$, $0 < z < 1$.[13]

When we have $K = 1$, when called by learner 1 in its explorations, learner 2 may always choose its suboptimal arm b since it is under-explored for learner 2. If this happens, then in exploitations learner 1 will almost always choose its own arm instead of learner 2, because it had estimated the accuracy of learner 2 for x^* incorrectly because the random rewards in explorations of learner 2 came from b. By letting $K \geq 1$, we also consider cases where only a fraction of reward samples of learner 2 for learner 1 comes from the suboptimal arm b. We will show that for any value of $K \geq 1$, there exists a problem instance of the form given in (22) such that learner 1's regret is linear in time. Let E_t be the event that time t is an exploitation slot for learner 1. Let $\hat{\pi}_m(t)$, $\hat{\pi}_2(t)$ be the sample mean reward of arm m and learner 2 for learner 1 at time t respectively. Let ξ_τ be the event that learner 1 exploits for the τth time by choosing its own arm. Denote the time of the τth exploitation of learner 1 by $\tau(t)$. We will show that for any finite τ, $\mathrm{P}(\xi_\tau, \ldots, \xi_1) \geq 1/2$. We have by the chain rule

$$\mathrm{P}(\xi_\tau, \ldots, \xi_1) = \mathrm{P}(\xi_\tau | \xi_{\tau-1}, \ldots, \xi_1) \mathrm{Pr}(\xi_{\tau-1} | \xi_{\tau-2}, \ldots, \xi_1) \ldots \mathrm{P}(\xi_1). \tag{23}$$

We will continue by bounding $\mathrm{P}(\xi_\tau | \xi_{\tau-1}, \ldots, \xi_1)$. When the event $E_{\tau(t)} \cap \xi_{\tau-1} \cap \cdots \cap \xi_1$ happens, we know that at least $\lceil \tau(t)^z \log \tau(t)/K \rceil$ of $\lceil \tau(t)^z \log \tau(t) \rceil$ reward samples of learner 2 for learner 1 comes from b. Let $A_t := \{\hat{\pi}_m(t) > \pi_m - \epsilon_1\}$, $B_t := \{\hat{\pi}_2(t) < \pi_g - \epsilon_2\}$ and $C_t := \{\hat{\pi}_2(t) < \hat{\pi}_m(t)\}$, for $\epsilon_1 > 0, \epsilon_2 > 0$. Given $\epsilon_2 \geq \epsilon_1 + 2\delta$, we have $(A_t \cap B_t) \subset C_t$. Consider the event $\{A_t^C, E_t\}$. Since on E_t, learner 1 selected m at least $t^z \log t$ times (given that z is large enough such that the reward estimate of learner 1's own arm is accurate), we have $\mathrm{P}(A_t^C, E_t) \leq 1/(2t^2)$, using a Chernoff bound. Let $N_g(t)$ ($N_b(t)$) be the number of times learner 2 has chosen arm g (b) when called by learner 1 by time t. Let $r_g(t)$ ($r_b(t)$) be the random reward of arm g (b) when it is

[12]Here $D_i(t)$ is the control function that controls when to explore or exploit the choices in $\mathcal{K}_i$ for learner i.

[13]Given two control functions of the form $C_i t^z \log t$, $i \in \{1, 2\}$, we can always normalize them such that one of them is $t^z \log t$ and the other one is $t^z \log t / K$, and then construct the problem instance that gives linear regret based on the normalized control functions.

chosen for the tth time by learner 2. For $\eta_1 > 0$, $\eta_2 > 0$, let $Z_1(t) := \{(\sum_{t'=1}^{N_g(t)} r_g(t'))/N_g(t) < \pi_g + \eta_1\}$ and $Z_2(t) := \{(\sum_{t'=1}^{N_b(t)} r_b(t'))/N_b(t) < \pi_b + \eta_2\}$. On the event $E_{\tau(t)} \cap \xi_{\tau-1} \cap \cdots \cap \xi_1$, we have $N_g(\tau(t))/N_b(\tau(t)) \leq K$. Since $\hat{\pi}_2(t) = \left(\sum_{t'=1}^{N_b(t)} r_b(t') + \sum_{t'=1}^{N_g(t)} r_g(t')\right)/(N_b(t) + N_g(t))$, We have

$$Z_1(t) \cap Z_2(t) \Rightarrow \hat{\pi}_2(t) < \frac{N_g(t)\pi_g + N_b(t)\pi_b + \eta_1 N_g(t) + \eta_2 N_b(t)}{N_b(t) + N_g(t)}. \quad (24)$$

If

$$\pi_g - \pi_b > \frac{N_g(t)}{N_b(t)}(\eta_1 + \epsilon_2) + (\eta_2 + \epsilon_2) \quad (25)$$

then, it can be shown that the right hand side of (24) is less than $\pi_g - \epsilon_2$. Thus given that (25) holds, we have $Z_1(t) \cap Z_2(t) \subset B_t$. But on the event $E_{\tau(t)} \cap \xi_{\tau-1} \cap \cdots \cap \xi_1$, (25) holds at $\tau(t)$ when $\pi_g - \pi_b > K(\eta_1 + \epsilon_2) + (\eta_2 + \epsilon_2)$. Note that if we take $\epsilon_1 = \eta_1 = \eta_2 = \delta/2$, and $\epsilon_2 = \epsilon_1 + 2\delta = 5\delta/2$ the statement above holds for a problem instance with $C_K > 3K + 3$. Since at any exploitation slot t, at least $\lceil t^z \log t/K \rceil$ samples are taken by learner 2 from both arms b and g, we have $\mathrm{P}(Z_1(\tau(t))^C) \leq 1/(4\tau(t)^2)$ and $\mathrm{P}(Z_2(\tau(t))^C) \leq 1/(4\tau(t)^2)$ by a Chernoff bound (again for z large enough as in the proofs of Theorems 1 and 2). Thus $\mathrm{P}(B_{\tau(t)})^C \leq \mathrm{P}(Z_1(\tau(t))^C) + \mathrm{P}(Z_2(\tau(t))^C) \leq 1/(2\tau(t)^2)$. Hence $\mathrm{P}(C^C_{\tau(t)}) \leq \mathrm{P}(A^C_{\tau(t)}) + \mathrm{P}(B^C_{\tau(t)}) \leq 1/(\tau(t)^2)$, and $\mathrm{P}(C_{\tau(t)}) > 1 - 1/(\tau(t)^2)$. Continuing from (23), we have

$$\begin{aligned}\mathrm{P}(\xi_\tau, \ldots, \xi_1) &= \left(1 - 1/(\tau(t)^2)\right)\left(1 - 1/((\tau-1)(t)^2)\right) \\ &\quad \cdots \left(1 - 1/((1)(t)^2)\right) \\ &\geq \Pi_{t'=2}^{\tau(t)}\left(1 - 1/(t')^2\right) > 1/2\end{aligned}$$

for all τ. This result implies that with probability greater than one half, learner 1 chooses its own arm at all of its exploitation slots, resulting in an expected per-slot regret of $\pi_g - \pi_m > \delta$. Hence the regret is linear in time. ■

B. Comparison of CLUP and DCZA

In this subsection we assess the computation and memory requirements of DCZA and compare it with CLUP. DCZA needs to keep the sample mean reward estimates of K_i choices for each active hypercube. A level l active hypercube becomes inactive if the context arrivals to that hypercube exceeds $2^{\rho l}$. Because of this, the number of active hypercubes at any time T may be much smaller than the number of activated hypercubes by time T. In the best-case, only one level l hypercube experiences context arrivals, then when that hypercube is divided into level $l+1$ hypercubes, only one of these hypercubes experiences context arrivals and so on. In this case, DCZA run with $\rho = 3\alpha$ creates at most $1 + (\log_2 T)/(3\alpha)$ hypercubes (using Lemma 5). In the worst-case (given in Corollary 1), DCZA creates at most $2^{2D}T^{D/(3\alpha+D)}$ hypercubes. Recall that for any D and α, the number of hypercubes of CLUP creates is $O(T^{D/(3\alpha+D)})$. Hence, in practice the memory requirement of DCZA can be much smaller than CLUP which requires to keep the estimates for every hypercube at all times. Finally DCZA does not require final time T as in input while CLUP requires it. Although CLUP can be combined with the doubling trick to make it independent of T, this makes the constants that multiply the time order of the regret large.

VII. Conclusion

In this paper we proposed a novel framework for decentralized, online learning by many learners. We developed two novel online learning algorithms for this problem and proved sublinear regret results for our algorithms. We discussed some implementation issues such as complexity and the memory requirement under different instance and context arrivals. Our theoretical framework can be applied to many practical settings including distributed online learning in Big Data mining, recommendation systems and surveillance applications. Cooperative contextual bandits opens a new research direction in online learning and raises many interesting questions: What are the lower bounds on the regret? Is there a gap in the time order of the lower bound compared to centralized contextual bandits due to informational asymmetries? Can regret bounds be proved when cost of calling learner j is controlled by learner j? In other words, what happens when a learner wants to maximize both the total reward from its own contexts and the total reward from the calls of other learners.

Appendix A
A Bound on Divergent Series

For $\rho > 0$, $\rho \neq 1$, $\sum_{t=1}^{T} 1/(t^\rho) \leq 1 + (T^{1-\rho} - 1)/(1-\rho)$.

Proof: See [23]. ■

Appendix B
Frequently Used Expressions

Mathematical Operators:

- $O(\cdot)$: Big O notation.
- $\tilde{O}(\cdot)$: Big O notation with logarithmic terms hidden.
- $\mathrm{I}(A)$: indicator function of event A.
- A^c or A^C: complement of set A.

Notation Related to Underlying System:

- $\mathcal{M}$: Set of learners. $M = |\mathcal{M}|$.
- $\mathcal{F}_i$: Set of arms of learner i. $F_i = |\mathcal{F}_i|$.
- $\mathcal{M}_{-i}$: Set of learners except i. $M_i = |\mathcal{M}_{-i}|$.
- $\mathcal{K}_i$: Set of choices of learner i. $K_i = |\mathcal{K}_i|$.
- $\mathcal{F}$: Set of all arms.
- $\mathcal{X} = [0,1]^D$: Context space.
- D: Dimension of the context space.
- $\pi_f(x)$: Expected reward of arm $f \in \mathcal{F}$ for context x.
- $\pi_j(x)$: Expected reward of learner j's best arm for context x.
- d^i_k: Cost of selecting choice $k \in \mathcal{K}_i$ for learner i.
- $\mu^i_k(x) = \pi_k(x) - d^i_k$: Expected net reward of learner i from choice k for context x.
- $k^*_i(x)$: Best choice (highest expected net reward) for learner i for context x.
- $f^*_i(x)$: Best arm (highest expected reward) of learner j for context x.
- L: Hölder constant. α: Hölder exponent.

Notation Related to Algorithms:

- $D_1(t), D_2(t), D_3(t)$: Control functions.
- p: Index for set of contexts (hypercube).
- m_T: Number of slices for each dimension of the context for CLUP.
- $\mathcal{P}_T$: Partition of $\mathcal{X}$ for CLUP.

- $\mathcal{P}_i(t)$: Learner i's adaptive partition of $\mathcal{X}$ at time t for DCZA.
- $\mathcal{P}(t)$: Union of partitions of $\mathcal{X}$ of all learners for DCZA.
- $p_i(t)$: The set in $\mathcal{P}_i(t)$ that contains $x_i(t)$.
- $\mathcal{M}_{i,p}^{\mathrm{uc}}(t)$: Set of learners who are training candidates of learner i at time t for set p of learner i's partition.
- $\mathcal{M}_{i,p}^{\mathrm{ut}}(t)$: Set of learners who are under-trained by learner i at time t for set p of learner i's partition.
- $\mathcal{M}_{i,p}^{\mathrm{ue}}(t)$: Set of learners who are under-explored by learner i at time t for set p of learner i's partition.
- $\mathcal{M}_{i,p}^{\mathrm{uc}}(t)$: Set of learners who are training candidates of learner i at time t for set p of learner i's partition.

REFERENCES

[1] K. Liu and Q. Zhao, "Distributed learning in multi-armed bandit with multiple players," *IEEE Trans. Signal Process.*, vol. 58, no. 11, pp. 5667–5681, 2010.

[2] C. Tekin and M. Liu, "Online learning in decentralized multi-user spectrum access with synchronized explorations," in *Proc. IEEE MILCOM*, 2012, pp. 1–6.

[3] R. Kleinberg, A. Slivkins, and E. Upfal, "Multi-armed bandits in metric spaces," in *Proc. 40th Annu. ACM Symp. Theory Comput.*, 2008, pp. 681–690.

[4] S. Bubeck, R. Munos, G. Stoltz, and C. Szepesvari, "X-armed bandits," *J. Mach. Learn. Res.*, vol. 12, pp. 1655–1695, 2011.

[5] A. Slivkins, "Contextual bandits with similarity information," in *Proc. 24th Annu. Conf. Learn. Theory (COLT)*, Jun. 2011, vol. 19, pp. 679–702.

[6] M. Dudik, D. Hsu, S. Kale, N. Karampatziakis, J. Langford, L. Reyzin, and T. Zhang, "Efficient optimal learning for contextual bandits," 2011, ArXiv preprint arXiv:1106.2369 [Online]. Available: http://arxiv.org/abs/1106.2369

[7] J. Langford and T. Zhang, "The epoch-greedy algorithm for contextual multi-armed bandits," *Adv. Neural Inf. Process. Syst.*, vol. 20, pp. 1096–1103, 2007.

[8] W. Chu, L. Li, L. Reyzin, and R. E. Schapire, "Contextual bandits with linear payoff functions," in *Proc. 14th Int. Conf. Artif. Intell. Statist. (AISTATS)*, Apr. 2011, vol. 15, pp. 208–214.

[9] L. Li, W. Chu, J. Langford, and R. E. Schapire, "A contextual-bandit approach to personalized news article recommendation," in *Proc. 19th Int. Conf. World Wide Web*, 2010, pp. 661–670.

[10] P. Auer, N. Cesa-Bianchi, and P. Fischer, "Finite-time analysis of the multiarmed bandit problem," *Mach. Learn.*, vol. 47, pp. 235–256, 2002.

[11] K. Crammer and C. Gentile, "Multiclass classification with bandit feedback using adaptive regularization," *Mach. Learn.*, vol. 90, no. 3, pp. 347–383, 2013.

[12] A. Anandkumar, N. Michael, and A. Tang, "Opportunistic spectrum access with multiple players: Learning under competition," in *Proc. IEEE INFOCOM*, Mar. 2010.

[13] C. Tekin and M. Liu, "Online learning of rested and restless bandits," *IEEE Trans. Inf. Theory*, vol. 58, no. 8, pp. 5588–5611, 2012.

[14] H. Liu, K. Liu, and Q. Zhao, "Learning in a changing world: Restless multiarmed bandit with unknown dynamics," *IEEE Trans. Inf. Theory*, vol. 59, no. 3, pp. 1902–1916, 2013.

[15] R. Stranders, L. Tran-Thanh, F. M. D. Fave, A. Rogers, and N. R. Jennings, "DCOPs and bandits: exploration and exploitation in decentralised coordination," in *Proc. 11th Int. Conf. Autonom. Agents Multiagent Syst.—Volume 1*, 2012, pp. 289–296.

[16] Y. Gai, B. Krishnamachari, and R. Jain, "Combinatorial network optimization with unknown variables: multi-armed bandits with linear rewards and individual observations," *IEEE/ACM Trans. Netw.*, vol. 20, no. 5, pp. 1466–1478, 2012.

[17] S. S. Ram, A. Nedic, and V. V. Veeravalli, "Distributed stochastic subgradient projection algorithms for convex optimization," *J. Optim. Theory Appl.*, vol. 147, no. 3, pp. 516–545, 2010.

[18] F. Yan, S. Sundaram, S. Vishwanathan, and Y. Qi, "Distributed autonomous online learning: regrets and intrinsic privacy-preserving properties," *IEEE Trans. Knowl. Data Eng.*, vol. 25, no. 11, pp. 2483–2493, 2013.

[19] M. Raginsky, N. Kiarashi, and R. Willett, "Decentralized online convex programming with local information," in *Proc. Amer. Control Conf. (ACC)*, 2011, pp. 5363–5369.

[20] C. Tekin, S. Zhang, and M. van der Schaar, "Distributed online learning in social recommender systems," *IEEE J. Sel. Topics Signal Process*, vol. 8, no. 4, pp. 638–652, Aug. 2014.

[21] H. Liu, K. Liu, and Q. Zhao, "Learning in a changing world: Non-Bayesian restless multi-armed bandit," Univ. of California—Davis, Tech. Rep., 2010.

[22] R. Ortner, "Exploiting similarity information in reinforcement learning," in *Proc. 2nd ICAART*, 2010, pp. 203–210.

[23] E. Chlebus, "An approximate formula for a partial sum of the divergent p-series," *Appl. Math. Lett.*, vol. 22, no. 5, pp. 732–737, 2009.

Cem Tekin (M'13) received the B.Sc. degree in electrical and electronics engineering from the Middle East Technical University, Ankara, Turkey, in 2008, the M.S.E. degree in electrical engineering: systems, M.S. degree in mathematics, Ph.D. degree in electrical engineering: systems from the University of Michigan, Ann Arbor, in 2010, 2011 and 2013, respectively. He is an Assistant Professor in Electrical and Electronics Engineering Department at Bilkent University, Turkey. From February 2013 to January 2015, he was a Postdoctoral Scholar at University of California, Los Angeles. His research interests include machine learning, multi-armed bandit problems, data mining, multi-agent systems and game theory. He received the University of Michigan Electrical Engineering Departmental Fellowship in 2008, and the Fred W. Ellersick award for the best paper in MILCOM 2009.

Mihaela van der Schaar (F'10) is Chancellor Professor of Electrical Engineering at University of California, Los Angeles. Her research interests include network economics and game theory, online learning, dynamic multi-user networking and communication, multimedia processing and systems, real-time stream mining. She is an IEEE Fellow, a Distinguished Lecturer of the Communications Society for 2011–2012, the Editor in Chief of IEEE TRANSACTIONS ON MULTIMEDIA and a member of the Editorial Board of the IEEE JOURNAL ON SELECTED TOPICS IN SIGNAL PROCESSING. She received an NSF CAREER Award (2004), the Best Paper Award from IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY (2005), the Okawa Foundation Award (2006), the IBM Faculty Award (2005, 2007, 2008), the Most Cited Paper Award from *EURASIP: Image Communications Journal* (2006), the Gamenets Conference Best Paper Award (2011) and the 2011 IEEE Circuits and Systems Society Darlington Award Best Paper Award. She received three ISO awards for her contributions to the MPEG video compression and streaming international standardization activities, and holds 33 granted U.S. patents.