Distributed Online Learning in Social Recommender Systems

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

***Abstract*—In this paper, we consider decentralized sequential decision making in distributed online recommender systems, where items are recommended to users based on their search query as well as their specific background including history of bought items, gender and age, all of which comprise the context information of the user. In contrast to centralized recommender systems, in which there is a single centralized seller who has access to the complete inventory of items as well as the complete record of sales and user information, in decentralized recommender systems each seller/learner only has access to the inventory of items and user information for its own products and not the products and user information of other sellers, but can get commission if it sells an item of another seller. Therefore, the sellers must distributedly find out for an incoming user which items to recommend (from the set of own items or items of another seller), in order to maximize the revenue from own sales and commissions. We formulate this problem as a cooperative contextual bandit problem, analytically bound the performance of the sellers compared to the best recommendation strategy given the complete realization of user arrivals and the inventory of items, as well as the context-dependent purchase probabilities of each item, and verify our results via numerical examples on a distributed data set adapted based on Amazon data. We evaluate the dependence of the performance of a seller on the inventory of items the seller has, the number of connections it has with the other sellers, and the commissions which the seller gets by selling items of other sellers to its users.**

***Index Terms*—Collaborative learning, contextual bandits, distributed recommender systems, multi-agent online learning, regret.**

I. Introduction

ONE of the most powerful benefits of a social network is the ability for cooperation and coordination on a large scale over a wide range of different agents [1]. By forming a network, agents are able to share information and opportunities in a mutually beneficial fashion. For example, companies can collaborate to sell products, charities can work together to raise money, and a group of workers can help each other search for jobs. Through such cooperation, agents are able to attain much greater rewards than would be possible individually. But sustaining efficient cooperation can also prove extremely challenging. First, agents operate with only incomplete information, and must learn the environment parameters slowly over time. Second, agents are decentralized and thus uncertain about their neighbor's information and preferences. Finally, agents are selfish in the sense that, they don't want to reveal their inventory, observations and actions to other agents, unless they benefit. This paper produces a class of algorithms that effectively addresses all of these issues: at once allowing decentralized agents to take near-optimal actions in the face of incomplete information, while still incentivizing them to fully cooperate within the network.

The framework we consider is very broad and applicable to a wide range of social networking situations. We analyze a group of agents 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 to offer to that user, and the user will either reject or accept each item. These items can represent a variety of things, from a good that the agent is trying to sell, to a cause that the agent is trying to promote, to a photo that the agent is trying to circulate. In each application, the action of accepting or rejecting by the user will likewise have a distinct meaning. 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, 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.

We allow for cooperation in this network by letting each agent recommend items of neighboring agents to incoming users, in addition to its own items. Thus if the specific background of the incoming user makes it unlikely for him to accept any of the agent's items, the agent can instead recommend him some items from a neighboring agent with more attractive offerings. By trading referrals in this fashion, all of the agents that are connected together can benefit. To provide proper incentives, a commission will be paid to the recommending agent every time an item is accepted by a user from the recommended agent. When defined appropriately, this commission ensures that both sides will benefit each time a recommendation occurs and thus is able to sustain cooperation.

However, since agents are decentralized, they do not directly share the information that they learn over time about user preferences for their own items. So when the decision to recommend a neighboring agent occurs, it is done based solely on

Manuscript received September 15, 2013; accepted December 16, 2013. Date of publication January 10, 2014; date of current version July 16, 2014. This work was supported in part by Grants NSF CNS 1016081 and AFOSR DDDAS. The guest editor coordinating the review of this manuscript and approving it for publication was Prof. Vikram Krishnamurthy.

C. Tekin and M. van der Schaar are with the Department of Electrical Engineering, University of California, Los Angeles, Los Angeles, CA 90095 USA. (e-mail: cmtkn@ucla.edu; mihaela@ee.ucla.edu).

S. Zhang is with the Department of Economics, University of California, Los Angeles, Los Angeles, CA 90095 USA (e-mail: simpsonzhang@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/JSTSP.2014.2299517

the previous successes the agent had when recommending that neighbor. Thus agents must learn about their neighbor's acceptance probabilities through their own trial and error, unlike in other social learning papers such as [2]–[5], where agents share information directly with their neighbors.

Another key feature of our algorithms is that they are non-Bayesian unlike [2], [3]. Instead we model the learning through contextual bandits, where the context is based on the user's background. By building upon the theory of contextual bandits, we produce a class of algorithms that allows agents to take near-optimal actions even with decentralized learning. We prove specific bounds for the regret, which is the difference between the total expected reward of an agent using a learning algorithm and the total expected reward of the optimal policy for the agent, which is computed given perfect knowledge about acceptance probabilities for each context. We show that the regret is sublinear in time in all cases. We further show that our algorithms can operate regardless of the specific network topology, including the degree of connectivity, degree distribution, clustering coefficient, etc., although the performance is better if the network is more connected since each agent will have access to more items of other agents.

The rest of the paper is organized as follows. Related work is given in Section II. The problem formulation is given in Section III. In Section IV, we consider the online learning problem involving multiple decentralized agents and analyze its regret. In Section IV-A we develop an algorithm to achieve sublinear regret when the purchase probability of the items depend on the other recommended items, and in Section IV-C we develop a faster learning algorithm when item purchase probabilities are independent of each other. The effect of connectivity between the agents is given in Section V. In Section VI, numerical results demonstrating the effects of commissions, size of the set of items of agents and connectivity of agents are given, using an artificial data set which is based on a real data set. Finally, we conclude the paper in Section VII.

II. Related Work

This work represents a significant departure from the other works in contextual bandits, which consider only centralized agents, single arms played at once, and no incentive issues. Most of the prior work on contextual bandits is focused on a single agent choosing one arm at a time based on the context information provided to it at each time slot [6]–[9]. In these works the system is centralized, so the agent can directly access all the arms. Our framework in this paper differs from the centralized contextual bandit framework in two important aspects. First, multiple agents who can only access a subset of arms, and who only get feedback about this subset, cooperate to maximize their total reward. Second, each agent can choose multiple arms at each time slot, which makes the arm selection problem combinatorial in the number of arms. To the best of our knowledge, our work is the first to provide rigorous solutions for online learning by multiple cooperative agents selecting multiple arms at each time step when context information is present. We had previously proposed a multi-agent contextual bandit framework in [10] where each agent only selects a single arm at each time slot. Different from this work, in this paper we assume that an agent can select multiple arms, and the expected reward of the agent from an arm may depend on the other selected arms. This makes the problem size grow combinatorially in the arm space, which requires the design of novel learning algorithms to quicken the learning process. Combinatorial bandits [11] have been studied before in the multi-armed bandit setting, but to the best of our knowledge we are the first to propose the decentralized contextual combinatorial bandit model studied in this paper. This decentralization is important because it allows us to analyze a social network framework and the fundamental challenges associated with it including commissions, third-party sellers, etc. We are also able to consider the specific effects of the network structure on the regret in our model. In contrast, our approach in [10] does not address the network structure concerns. Several other examples of related work in contextual bandits are [12], in which a contextual bandit model is used for recommending personalized news articles based on a variant of the UCB1 algorithm in [13] designed for linear rewards, and [14] in which the authors solve a classification problem using contextual bandits, where they propose a perceptron based algorithm that achieves sublinear regret.

TABLE I
Comparison With Related Work in Multi-Armed Bandits

	[6]–[9]	[15], [18] [19]	[10]	This work
Multi-agent	no	yes	yes	yes
Cooperative	N/A	yes	yes	yes
Contextual	yes	no	yes	yes
Context arrival process	arbitrary	N/A	arbitrary	arbitraty
(syn)chronous, (asyn)chronous	N/A	syn	both	both
Regret	sublinear	log	sublinear	sublinear
Multi-play for each agent	no	no	no	yes
Action set size combinatorial in number of agents and items	no	no	no	yes
Action sets of the agents	N/A	same	different	different

Apart from contextual bandits, there is a large set of literature concerned in multi-user learning using a multi-armed bandit framework [15]–[19]. 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 bandits framework [6]. The main differences are that: (i) a three phase learning algorithm with *training*, *exploration* and *exploitation* phases is needed instead of the standard two phase algorithms with *exploration* and *exploitation* phases that are commonly used in centralized contextual bandit problems; (ii) the adaptive partitions of the context space should be formed in a way that each learner can efficiently utilize what is learned by other learners about the same context; (iii) since each agent has multiple selections at each time slot, the set of actions for each agent is very large, making the learning rate slow. Therefore, the correlation between the arms of the agents should be exploited to quicken the learning process. In our distributed multi-agent multiple-play contextual bandit formulation, the training phase, which balances the learning rates of the agents, is necessary

TABLE II
COMPARISON WITH PRIOR WORK IN RECOMMENDER SYSTEMS

	Item-based (IB), user-based (UB)	Memory-based, model-based	Uses context info.	Performance measure	Similarity distance	Centralized(C), Decentralized(D)
[30]	UB	Memory-based	No	Ranking precision	-	C
[22]	UB	Bayesian-based latent semantic model	No	MAE, RMS, 0/1 loss	Pearson correlation	C
[23]	UB	Bayesian-based Markov model	No	Precision& Recall	Pearson correlation	C
[24]	IB	Cluster model	No	-	Cosine	C
[25]	UB	Memory-based	Yes	Precision& Recall	-	C
[26]	UB	Bayesian classifier model	No	Precision& Recall	Pearson correlation	C
[27]	UB	Cluster model	No	MAE& Coverage	Pearson correlation	C
[28]	UB	MDP model	No	Recall	Self-defined similarity	C
[20]	UB	MAB model	No	Reward	Lipschitz continuous	C
[21]	UB	MAB model	Yes	Regret	Lipschitz continuous	C
Our work	UB	MAB model	Yes	Regret	Hölder continuous	D

since the context arrivals to agents are different which makes the learning rates of the agents for various context different.

There is also an extensive literature on recommender systems that incorporates a variety of different methods and frameworks. Table II provides a summary of how our work is related to other work. Of note, there are several papers that also use a similar multi-armed bandit framework for recommendations. For example, [20] considers a bandit framework where a recommender system learns about the preferences of users over time as each user submits ratings. It uses a linear bandit model for the ratings, which are assumed to be functions of the specific user as well as the features of the product. [21] is another work that utilizes multi-armed bandits in a recommendation system. It considers a model that must constantly update recommendations as both preferences and the item set changes over time.

There are also numerous examples of works that do not use a bandit framework for recommendations. One of the most commonly used methods for recommendations are collaborative filtering algorithms such as [22]–[28], which make recommendations by predicting the user's preferences based on a similarity measure with other users. Items with the highest similarity score are then recommended to each user; for instance items may be ranked based on the number of purchases by similar users. There are numerous ways to perform the similarity groupings, such as the cluster model in [24], [27] that groups users together with a set of like-minded users and then makes recommendations based on what the users in this set choose. Another possibility is presented in [25], which pre-filters ratings based on context before the recommendation algorithm is started.

An important difference to keep in mind is that the recommendation systems in other works are a single centralized system, such as Amazon or Netflix. Thus the system has complete information at every moment in time, and does not need to worry about incentive issues or pay commissions. However, in this paper each agent is in effect its own separate recommendation system, since agents do not directly share information with each other. Therefore the mechanism we propose must be applied separately by every agent in the system based on that agent's history of user acceptances. So in effect our model is a giant collection of recommendation systems that are running in parallel. The only cross-over between these different systems is when one agent suggests which of its items should be recommended by another agent. This allows for an indirect transfer of information, and lets that other agent make better choices than it could without this suggested list of items.

Also, it is important to note that decentralization in the context of our paper does not mean the same thing as in other papers such as [29]. Those papers assume that the $users$ are decentralized, and develop mechanisms based on trust and the similarity of different users in order to allow $users$ to recommend items to each other. We are assuming that the $agents$ are decentralized, and so each user still only gets a recommendation from one source, it is just that this source may be different depending on which agent this user arrives at. Thus this paper is fundamentally different from the works in that literature.

III. PROBLEM FORMULATION

There are M decentralized agents/learners which are indexed by the set $\mathcal{M} := \{1, 2, \ldots, M\}$. Let $\mathcal{M}_{-i} := \mathcal{M} - \{i\}$. Each agent i has an inventory of items denoted by $\mathcal{F}_i$, which it can offer to its users and the users of other agents by paying some commission. Users arrive to agents in a discrete time setting ($t = 1, 2, \ldots$). Agent i recommends N items to its user at each time slot. For example, N can be the number of recommendation slots the agent has on its website, or it can be the number of ads it can place in a magazine. We assume that N is fixed throughout the paper. Each item $f \in \mathcal{F}_i$ has a fixed price $p_f^i > 0$. We assume that the inventories of agents are mutually disjoint.[1] For now, we will assume that all the agents in the network are directly connected to each other, so that any agent can sell items to the users of any other agent directly without invoking intermediate agents. We will discuss the agents in more general network topologies in Section V. Let $\mathcal{F} := \cup_{i \in \mathcal{M}} \mathcal{F}_i$ be the set of items of all agents. We assume that there is an unlimited supply of each type of item. This assumption holds for digital goods such as e-books, movies, videos, songs, photos, etc. An agent does not know the inventory of items of the other agents but knows an upper bound on $|\mathcal{F}_j|$,[2] $j \in \mathcal{M}_{-i}$ which is equal to $F_{\max}$.

We note that our model is suitable for a broad range of applications. The agent can represent a company, an entrepreneur, a content provider, a blogger, etc., and the items can be goods, services, jobs, videos, songs, etc. The notion of an item can be generalized even further to include such things as celebrities that are recommended to be followed in Twitter, Google+, etc. And the prices that we introduce later can also be generalized to mean any type of benefit the agent can receive from a user. For expositional convenience, we will adopt the formulation of firms selling goods to customers for most of the paper, but we

[1]Even when an item is in the inventory of more than one agent, its price can be different among these agents, thus different IDs can be assigned to these items. We do not assume competition between the agents, hence our methods in this paper will work even when the inventories of agents are not mutually disjoint.

[2]For a set A, $|A|$ denotes its cardinality.

emphasize that many other interpretations are equally valid and applicable.

Notation

Natural logarithm is denoted by $\log(\cdot)$. For sets $\mathcal{A}$ and $\mathcal{B}$, $\mathcal{A}-\mathcal{B}$ denotes the elements of $\mathcal{A}$ that are not in $\mathcal{B}$. $P(\cdot)$ is the probability operator, $E[\cdot]$ is the expectation operator. For an algorithm α, $E_\alpha[\cdot]$ denotes the expectation with respect to the distribution induced by α. Let $[t] := \{1,\ldots,t-1\}$. Let $\beta_2 := \sum_{t=1}^{\infty} 1/t^2$.

Definition of Users With Contexts

At each time slot $t = 1, 2, \ldots$, a user with a specific search query indicating the type of item the user wants, or other information including a list of previous purchases, price-range, age, gender etc., arrives to agent i. We define all the properties of the arriving user known to agent i at time t as the context of that user, and denote it by $x_i(t)$. We assume that the contexts of all users belong to a known space $\mathcal{X}$, which without loss of generality is taken to be $[0,1]^d$ in this paper, where d is the dimension of the context space. Although the model we propose in this paper has synchronous arrivals, it can be easily extended to the asynchronous case where agents have different user arrival rates, and even when no user arrives in some time slots. The only difference of this from our framework is that instead of keeping the same time index t for every agent, we will have different time indices for each agent depending on the number of user arrivals to that agent.

Definition of Commissions

In order to incentivize the agents to recommend each other's items, they will provide commissions to each other. In this paper we focus on *sales commission*, which is paid by the recommended agent to the recommending agent every time a user from the recommending agent purchases an item of the recommended agent. We assume that these commissions are fixed at the beginning and do not change over time. The system model is shown in Fig. 1. Basically, if agent i recommends an item f of agent j to its user, and if that user buys the item of agent j, then agent i obtains a fixed commission which is equal to $c_{i,j} > 0$. All of our results in this paper will also hold for the case when the commission is a function of the price of the item f sold by agent j, i.e., $c_{i,j}(p_f^j)$. However we use the fixed commission assumption, since privacy concerns may cause the agent j or the user to not want to reveal the exact sales price to agent i. We assume that $c_{i,j} \leq p_f^j$ for all $i, j \in \mathcal{M}$, $f \in \mathcal{F}_j$. Otherwise, full cooperation can result in agent j obtaining less revenue than it would have obtained without cooperation. Nevertheless, even when $c_{i,j} > p_f^j$ for some $f \in \bar{\mathcal{F}}_j \subset \mathcal{F}_j$, by restricting the set of items that agent j can recommend to agent i to $\mathcal{F}_j - \bar{\mathcal{F}}_j$, all our analysis for agent i in this paper will hold.

Agents may also want to preserve the privacy of the items offered to the users, hence the privacy of their inventory in addition to the privacy of prices. This can be done with the addition of a *regulator* to the system. The regulator is a non-strategic entity whose goal is to promote privacy among the agents. If agent i wants to recommend an item from agent j, it can pass this request to the regulator and the regulator can create private keys for agent j and the user of agent i, such that agent j sends its recommendation to agent i's website in an encrypted format so that recommendation can only be viewed by the user, who has the key. The regulator can also control the transaction between agent j and agent i's user such that agent j pays its commission to agent i if the item is sold. Note that the regulator does not need to store the previous purchase histories or the inventories of the agents. Also it only regulates transactions between the agents but not the recommendations of an agent to its own users.

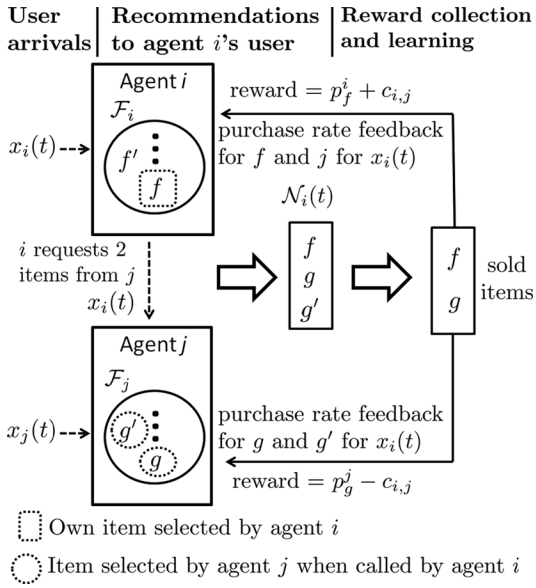


Fig. 1. Operation of the system for agent i for $N = 3$ recommendations. At each time a user arrives to agent i with context $x_i(t)$, agent i recommends a set of its own items and items from other agents, which is denoted by $\mathcal{N}_i(t)$.

A. Recommendations, Actions and Purchase Probabilities

The N items that agent i recommends to its user are chosen in the following way: An item can be chosen from the inventory $\mathcal{F}_i$ of agent i, or agent i can call another agent j and send the context information of $x_i(t)$ of its user, then agent j returns back an item f' with price $p_{f'}^j$,[3] to be recommended to agent i based on $x_i(t)$. Let $\mathcal{N}_i(t)$ be the set of items recommended by agent i to its user at time t. Let $\mathcal{A}_N$ be the set of subsets of $\mathcal{F}$ with N items. Let $\mathcal{N} \in \mathcal{A}_N$ be a set of N recommended items.

Consider agent i. Let $\mathcal{U}_i := \mathcal{M}_{-i} \cup \mathcal{F}_i$. For $u_i \in \mathcal{F}_i$, let $(u_i, 1)$ $((u_i, 0))$ denote the event that u_i is recommended (not recommended) to agent i's user. For $u_i \in \mathcal{M}_{-i}$, let (u_i, n_{u_i}) denote the event that agent i requests n_{u_i} (distinct) items to be recommended to its user from agent u_i. Let $(\boldsymbol{u}_i, \boldsymbol{n}_i) := \{(u_i, n_{u_i})\}_{u_i \in \mathcal{U}_i}$ be an action for agent i. Based on this, let

$$\mathcal{L}_i := \Bigg\{ (\boldsymbol{u}_i, \boldsymbol{n}_i) \text{ such that } n_{u_i} \in \times\{0,1\} \text{ for } u_i \in \mathcal{F}_i \text{ and } n_{u_i} \in \{0,\ldots,N\} \text{ for } u_i \in \mathcal{M}_{-i} \text{ and } \sum_{u_i \in \mathcal{U}_i} n_{u_i} = N \Bigg\},$$

be the set of actions available to agent i. We assume that $|\mathcal{F}_j| \geq N$ for all $j \in \mathcal{M}$. Let k_i be the components of vector $(\boldsymbol{u}_i, \boldsymbol{n}_i) \in \mathcal{L}_i$ with nonzero n_{u_i}. Since $n_{u_i} = 0$ means the corresponding

[3]Another method to preserve the privacy of prices is the following: Agent j's item will be recommended by agent i without a price tag, and when the user clicks to agent j's item it will be directed to agent j's website where the price will be revealed to the user.

item will not be recommended for $u_i \in \mathcal{F}_i$, and the corresponding agent will not recommend any items to agent i for $u_i \in \mathcal{M}_{-i}$, k_i is enough to completely characterize the action of agent i. Thus, for notational simplicity we denote an action of agent i by k_i. With an abuse of notation $k_i \in \mathcal{L}_i$ means that $(\boldsymbol{u}_i, \boldsymbol{n}_i)$ corresponding to k_i is in $\mathcal{L}_i$. For $u_i \in \mathcal{U}_i$, we let $n_{u_i}(k_i)$ be the value of n_{u_i} in the vector $(\boldsymbol{u}_i, \boldsymbol{n}_i)$ corresponding to k_i. Let $\mathcal{M}_i(k_i)$ be the set of agents in $\mathcal{M}_{-i}$ that recommend at least one item to agent i, when it chooses action k_i. We define $m_j(k_i)$ as the number of items agent j recommends to agent i when agent i chooses action $k_i \in \mathcal{L}_i$. Clearly we have $m_j(k_i) = n_j(k_i)$ for $j \in \mathcal{M}_{-i}$, and $m_i(k_i) = \sum_{u_i \in \mathcal{F}_i} n_{u_i}(k_i)$. Let $\boldsymbol{m}(k_i) := (m_1(k_i), \ldots, m_M(k_i))$ be the *recommendation vector* given action k_i,

$$\boldsymbol{m}_{-j}(k_i) := (m_1(k_i), \ldots, m_{j-1}(k_i), m_{j+1}(k_i), \ldots, m_M(k_i)),$$

and $\mathcal{F}_i(k_i) \subset \mathcal{F}_i$ be the set of own items recommended by agent i to its user when it takes action $k_i \in \mathcal{L}_i$. For a set of recommended items $\mathcal{N} \in \mathcal{A}_N$, let $\mathcal{G}_j(\mathcal{N}) := \mathcal{N} \cap \mathcal{F}_j$ and $g_j(\mathcal{N}) := |\mathcal{G}_j(\mathcal{N})|$. Let $\boldsymbol{g}_{-i}(\mathcal{N}) := \{g_j(\mathcal{N})\}_{j \in \mathcal{M}_{-i}}$. Below, we define the purchase probability of an item f that is recommended to a user with context x along with the set of items $\mathcal{N} \in \mathcal{A}_N$.

Assumption 1: **Group-dependent purchase probability:** We assume that the inventory of each agent forms a separate group (or type) of items. For example, if agents are firms, then firm i may sell televisions while firm j sells headphones. For each item f recommended together with the set of items $\mathcal{N} \in \mathcal{A}_N$, if f is an item of agent j, i.e., it belongs to group $\mathcal{F}_j$, then its acceptance probability will depend on other recommended items in the same group, i.e., $\mathcal{G}_j(\mathcal{N})$. The acceptance probability of an item $f \in \mathcal{F}_j$ may also depend on groups of other items offered together with f and also on their number, i.e., $\boldsymbol{g}_{-j}(\mathcal{N})$, but not on their identities. For $f \in \mathcal{F}_j$, a user with context $x \in \mathcal{X}$ will buy the item with an unknown probability $q_f(x, \mathcal{N}) := q_f(x, \mathcal{G}_j(\mathcal{N}), \boldsymbol{g}_{-j}(\mathcal{N}))$. For all $f \in \mathcal{F}$, users with contexts *similar* to each other have *similar* purchase probabilities. This similarity is characterized by a Hölder condition that is known by the agents, i.e., there exists $L > 0$, $\alpha > 0$ such that for all $x, x' \in \mathcal{X} = [0,1]^d$, we have

$$|q_f(x, \mathcal{N}) - q_f(x', \mathcal{N})| \leq L\|x - x'\|^\alpha,$$

where $\|\cdot\|$ denotes the Euclidian norm in $\mathbb{R}^d$.

In Assumption 1, we do not require the purchase probabilities to be similar for different sets of recommendations from the same group, i.e., for $f \in \mathcal{N}$ and $f \in \mathcal{N}'$ such that $\mathcal{G}_j(\mathcal{N}) \neq \mathcal{G}_j(\mathcal{N}')$, the purchase probability of f can be different for context $x \in \mathcal{X}$. Even though the Hölder condition can hold with different constants L_f and α_f for each item $f \in \mathcal{F}$, taking L to be the largest among L_f and α to be the smallest among α_f we get the condition in Assumption 1. For example, the context can be the (normalized) age of the user, and users with ages similar to each other can like similar items. We will also consider a simplified version, in which the purchase probability of an item is independent of the set of other items recommended along with that item.

Assumption 2: **Independent purchase probability:** For each item f offered along with the items in the set $\mathcal{N} \in \mathcal{A}_N$, a user with context x will buy the item with an unknown probability $q_f(x, \mathcal{N}) := q_f(x)$, independent of the other items in $\mathcal{N}$, for which there exists $L > 0$, $\alpha > 0$ such that for all $x, x' \in \mathcal{X}$, we have $|q_f(x) - q_f(x')| \leq L\|x - x'\|^\alpha$.

When Assumption 2 holds, the agents can estimate the purchase probability of an item by using the empirical mean of the number of times the item is purchased by users with similar context information. The purchase probabilities of items can be learned much faster in this case compared with Assumption 1, since the empirical mean will be updated every time the item is recommended. However, when the purchase probability of an item is group-dependent, the agents need to learn its purchase probability separately for each group the item is in.

B. Objective, Rewards and the Regret

The goal of agent i is to maximize its total expected reward (revenue). However, since it does not know the purchase probability of any item or the inventories of other agents a-priori, it must learn how to make optimal recommendations to its users over time. In this subsection we define the optimal actions for the agents and the "oracle" benchmark (optimal) solution the agents' learning algorithms compete with, which is derived from the unknown purchase probabilities and inventories.

The expected reward of agent i at time t from recommending a set of items $\mathcal{N}_i$ to its user with context $x_i(t) = x$ is given by

$$\sigma_i(\mathcal{N}_i, x) := \sum_{f \in (\mathcal{N}_i - \mathcal{F}_i)} c_{i,j(f)} q_f(x, \mathcal{N}_i) + \sum_{f \in \mathcal{N}_i \cap \mathcal{F}_i} p_f^i q_f(x, \mathcal{N}_i),$$

where $j(f)$ is the agent who owns item f. Based on this, given a user with context x, the set of items which maximizes the one-step expected reward of agent i is

$$\mathcal{N}_i^*(x) := \arg\max_{\mathcal{N} \in \mathcal{A}_N} (\sigma_i(\mathcal{N}, x)). \tag{1}$$

Since the inventory of other agents and $q_f(x, \mathcal{N})$, $x \in \mathcal{X}$, $\mathcal{N} \in \mathcal{A}_N$ are unknown a priori to agent i, $\mathcal{N}_i^*(x)$ is unknown to agent i for all contexts $x \in \mathcal{X}$. For a *recommendation vector* $\boldsymbol{m} = (m_1, \ldots, m_M)$, agent j has $\binom{|\mathcal{F}_j|}{m_j}$ actions $k_j \in \mathcal{L}_j$ for which $m_{j'}(k_j) = m_{j'}$ for all $j' \neq j$. Denote this set of actions of agent j by $\mathcal{B}_j(\boldsymbol{m})$. For an action $k_j \in \mathcal{B}_j(\boldsymbol{m})$, let

$$\pi_{j,k_j}(x) := \sum_{f \in \mathcal{F}_j(k_j)} q_f(x, \mathcal{F}_j(k_j), \boldsymbol{m}_{-j}(k_j)),$$

be the *purchase rate* of action k_j[4] for a user with context x. The *purchase rate* is equal to the expected number of items in $\mathcal{F}_j(k_j)$ sold to a user with context x, when the number of items recommended together with $\mathcal{F}_j(k_j)$ from the other groups is given by the vector $\boldsymbol{m}_{-j}(k_j)$. For actions of agent i such that $m_j(k_i) > 0$, the best set of items agent j can recommend to agent i (which maximizes agent i's expected commission from agent j) for a user with context x is $\mathcal{F}_j(k_j^*(k_i, x))$, where

$$k_j^*(k_i, x) := \arg\max_{k_j \in \mathcal{B}_j(\boldsymbol{m}(k_i))} \pi_{j,k_j}(x).$$

According to the above definitions, the expected reward of action $k_i \in \mathcal{L}_i$ to agent i for a user with context x is defined as

$$\mu_{i,k_i}(x) := \sum_{f \in \mathcal{F}_i(k_i)} p_f^i q_f(x, \mathcal{F}_i(k_i), \boldsymbol{m}_{-i}(k_i)) + \sum_{j \in \mathcal{M}_i(k_i)} c_{i,j} \pi_{j,k_j^*(k_i,x)}(x). \tag{2}$$

[4]If $\mathcal{F}_j(k_j) = \emptyset$, then $\pi_{j,k_j}(x) = 0$.

Then, given a user with context x, the best action of agent i is

$$k_i^*(x) := \arg\max_{k_i \in \mathcal{L}_i} \mu_{i,k_i}(x). \tag{3}$$

Note that for both Assumptions 1 and 2, $\mu_{i,k_i^*(x)}(x) = \sigma_i(\mathcal{N}_i^*(x), x)$. Let α_i be the recommendation strategy adopted by agent i for its own users, i.e., based on its past observations and decisions, agent i chooses a vector $\alpha_i(t) \in \mathcal{L}_i$ at each time slot. Let β_i be the recommendation strategy adopted by agent i when it is called by another agent to recommend its own items. Let $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_M)$ and $\boldsymbol{\beta} = (\beta_1, \ldots, \beta_M)$. Let $Q^i_{\boldsymbol{\alpha},\boldsymbol{\beta}}(T)$ be the expected total reward of agent i by time T from item sales to its own users and users of other agents and commissions it gets from item sales of other agents to its own users. For $f \in \mathcal{F}_i$, let $Y_f^{\alpha_i}(t)$ be a random variable which is equal to 1 when agent i recommends its own item f to its user at time t (0 otherwise), let $Y_f^{\beta_i,\alpha_j}(t)$ be a random variable which is equal to 1 when agent i recommends its own item f to agent j when it is called by agent j at time t (0 otherwise). For $j \in \mathcal{M}_{-i}$, let $Y_j^{\alpha_i}(t)$ be a random variable that is equal to 1 when agent i asks for recommendations from agent j at time t for its own user (0 otherwise). Let the (random) reward agent i gets from the set of recommendations made by $(\boldsymbol{\alpha}, \boldsymbol{\beta})$ to its user at time t be

$$O^i_{\boldsymbol{\alpha},\boldsymbol{\beta}}(t, x_i(t)) := \sum_{f \in \mathcal{F}_i} Y_f^{\alpha_i}(t) p_f^i q_f(x_i(t), \mathcal{N}_i(t)) + \sum_{j \in \mathcal{M}_{-i}} Y_j^{\alpha_i}(t) \left(\sum_{f \in \mathcal{F}_j} Y_f^{\beta_j,\alpha_i}(t) c_{i,j} q_f(x_i(t), \mathcal{N}_i(t)) \right), \tag{4}$$

and let

$$S^i_{\boldsymbol{\alpha},\boldsymbol{\beta}}(T) := E_{\boldsymbol{\alpha},\boldsymbol{\beta}} \left[\sum_{t=1}^{T} O^i_{\boldsymbol{\alpha},\boldsymbol{\beta}}(t, x_i(t)) \right],$$

be the total expected reward agent i can get based only on recommendations to its own users by time T. Let

$$U^i_{\boldsymbol{\alpha},\boldsymbol{\beta}}(t, x_j(t)) := Y_i^{\alpha_j}(t) \sum_{f \in \mathcal{F}_i} Y_f^{\beta_i,\alpha_j}(t), \tag{5}$$

be the number of items of agent i purchased by agent j's user at time t. Clearly we have

$$Q^i_{\boldsymbol{\alpha},\boldsymbol{\beta}}(T) = S^i_{\boldsymbol{\alpha},\boldsymbol{\beta}}(T) + E_{\boldsymbol{\alpha},\boldsymbol{\beta}} \left[\sum_{t=1}^{T} \sum_{j \in \mathcal{M}_{-i}} c_{i,j} U^i_{\boldsymbol{\alpha},\boldsymbol{\beta}}(t, x_j(t)) \right].$$

We can see that $Q^i_{\boldsymbol{\alpha},\boldsymbol{\beta}}(T) \geq S^i_{\boldsymbol{\alpha},\boldsymbol{\beta}}(T)$. Agent i's goal is to maximize its total reward $S^i_{\boldsymbol{\alpha},\boldsymbol{\beta}}(T)$ from its own users for any T. Since agents are cooperative, agent i also helps other agents $j \in \mathcal{M}_{-i}$ to maximize $S^j_{\boldsymbol{\alpha},\boldsymbol{\beta}}(T)$ by recommending its items to them. Hence the total reward the agent gets, $Q^i_{\boldsymbol{\alpha},\boldsymbol{\beta}}(T)$, is at least $S^i_{\boldsymbol{\alpha},\boldsymbol{\beta}}(T)$.

We assume that user arrivals to the agents are independent of each other. Therefore, agent j will also benefit from agent i if its item can be sold by agent i. In this paper, we develop distributed online learning algorithms $(\boldsymbol{\alpha}, \boldsymbol{\beta})$ for the agents in $\mathcal{M}$ such that the expected total reward $S^i_{\boldsymbol{\alpha},\boldsymbol{\beta}}(T)$ for any agent $i \in \mathcal{M}$ is maximized. This corresponds to minimizing the regret, which is given for agent i at time T as

$$R^i_{\boldsymbol{\alpha},\boldsymbol{\beta}}(T) := \sum_{t=1}^{T} \mu_{i,k_i^*(x_i(t))}(x_i(t)) - S^i_{\boldsymbol{\alpha},\boldsymbol{\beta}}(T). \tag{6}$$

Note that the regret is calculated with respect to the highest expected reward agent i can obtain from its own users, but not the users of other agents. Therefore, agent i does not act strategically to attract the users of other agents, such as by cutting its own prices or paying commissions even when an item is not sold to increase its chance of being recommended by another agent. We assume that agents are fully cooperative and follow the rules of the proposed algorithms. We provide a comparison between cooperative and strategic agents in the following remark.

Remark 1: In our cooperative algorithms, an agent j when called by agent i recommends a set of items that has the highest probability of being purchased by agent i's user. This recommendation does not decrease the reward of agent j, compared to the case when agent j does not cooperate with any other agent, since we assume that $p_f^j \geq c_{i,j}$ for all $f \in \mathcal{F}_j$. However, when the commission is fixed, recommending the set of items with the highest purchase rate does not always maximize agent j's reward. For example, agent j may have another item which has a slightly smaller probability of being purchased by agent i's user, but has a much higher price than the item which maximizes the purchase rate. Then, it is of interest to agent j to recommend that item to agent i rather than the item that maximizes the purchase rate. This problem can be avoided by charging a commission which is equal to a percentage of the sales price of the item, i.e., $c_{i,j}(p_f^j) = c_{i,j} p_f^j$ for some $0 < c_{i,j} < 1$ for $f \in \mathcal{F}_j$. Our theoretical results will hold for this case as well. We will numerically compare the effects of a fixed commission and a commission which is equal to a percentage of the sales price on the learning rates and set of items recommended by the agents in Section VI.

We will show that the regret of the algorithms proposed in this paper will be sublinear in time, i.e., $R^i_{\boldsymbol{\alpha},\boldsymbol{\beta}}(T) = O(T^\phi)$, $\phi < 1$, which means that the distributed learning scheme converges to the average reward of the best recommender strategy $\mathcal{N}_i^*(x)$ given in (1) (or equivalently $k_i^*(x)$ given in (3)) for each $i \in \mathcal{M}$, $x \in \mathcal{X}$, i.e., $R^i_{\boldsymbol{\alpha},\boldsymbol{\beta}}(T)/T \to 0$. Moreover, the regret also provides us with a bound on how fast our algorithm converges to the best recommender strategy.

IV. Contextual Partitioning Algorithms for Multiple Recommendations

In this section we propose a series of distributed online learning algorithms called *Context Based Multiple Recommendations* (CBMR). We denote the part of CBMR which gives an action to agent i at every time slot for its own user with α_i^{CBMR}, and the part of CBMR which gives an action to agent i at every time slot for the recommendation request of another agent with β_i^{CBMR}. When clear from the context, we will drop the superscript. Basically, an agent using CBMR forms a partition of the context space $[0,1]^d$, depending on the final time T, consisting of $(m_T)^d$ sets where each set is a d-dimensional hypercube with dimensions $1/m_T \times 1/m_T \times \cdots \times 1/m_T$. The sets in this partition are indexed by $\mathcal{I}_T = \{1, 2, \ldots, (m_T)^d\}$.

Context Based Multiple Recommendations for dependent purchase probabilities (CBMR-d for agent i):

Input: $D_1(t)$, $D_{2,k}(t)$, $k \in \tilde{\mathcal{L}}_i$, $D_3(t)$, T, m_T
Initialize: Partition $[0,1]^d$ into $(m_T)^d$ sets, indexed by the set $\mathcal{I}_T = \{1, 2, \ldots, (m_T)^d\}$.
$N^i_{k,l} = 0, \forall k \in \mathcal{L}_i, l \in \mathcal{I}_T$, $N^i_{1,k,l} = 0$, $N^i_{2,k,l} = 0$, $\forall k \in \tilde{\mathcal{L}}_i, l \in \mathcal{I}_T$.
$\boldsymbol{N}^i_l = \{N^i_{k,l}\}_{k \in \mathcal{L}_i}$, $\boldsymbol{N}^i_{2,l} = \{N^i_{2,k,l}\}_{k \in \tilde{\mathcal{L}}_i}$.
$\bar{r}^i_{k,l} = 0$, $\bar{\pi}^i_{k,l} = 0$, for all $k \in \mathcal{L}_i, I_l \in \mathcal{I}_T$.
$\bar{\boldsymbol{r}}^i_l = \{\bar{r}^i_{k,l}\}_{l \in \mathcal{L}_i}$, $\bar{\boldsymbol{\pi}}^i_l = \{\bar{\pi}^i_{k,l}\}_{l \in \mathcal{L}_i}$.
while $t \geq 1$ **do**
  Algorithm α_i (Send recommendations to own users)
  for $l = 1, \ldots, (m_T)^d$ **do**
    if $x_i(t) \in I_l$ **then**
      if $\exists k \in \hat{\mathcal{L}}_i$ such that $N^i_{k,l} \leq D_1(t)$ **then**
        Run **Explore**(k, $N^i_{k,l}$, $\bar{r}^i_{k,l}$, t)
      else if $\exists k \in \tilde{\mathcal{L}}_i$ such that $N^i_{1,k,l} \leq D_{2,k}(t)$ **then**
        Run **Train**(k, $N^i_{1,k,l}$, t)
      else if $\exists k \in \tilde{\mathcal{L}}_i$ such that $N^i_{2,k,l} \leq D_3(t)$ **then**
        Run **Explore**(k, $N^i_{2,k,l}$, $\bar{r}^i_{k,l}$, t)
      else
        Run **Exploit**($\boldsymbol{N}^i_l, \boldsymbol{N}^i_{2,l}, \bar{\boldsymbol{r}}^i_l$, t)
      end if
    end if
  end for
  Algorithm β_i (Send recommendations to other agents' users)
  for $j \in \mathcal{M}_{-i}$ such that $i \in \mathcal{M}_j(\alpha_j(t))$ **do**
    Receive $x_j(t)$ and $\boldsymbol{m}(\alpha_j(t))$
    for $l = 1, \ldots, (m_T)^d$ **do**
      if $x_j(t) \in I_l$ **then**
        if $\exists\ k \in \mathcal{B}_i(\boldsymbol{m}(\alpha_j(t)))$ such that $N^i_{k,l} \leq D_1(t)$ **then**
          Run **Explore2**(k, $N^i_{k,l}$, $\bar{\pi}^i_{k,l}$, j, t)
        else
          Run **Exploit2**($\boldsymbol{N}^i_l$, $\bar{\boldsymbol{\pi}}^i_l$, $\boldsymbol{m}(\alpha_j(t))$, j, t)
        end if
      end if
    end for
  end for
  $t = t + 1$
end while

Fig. 2. Pseudocode of CBMR-d for agent i.

We denote the set with index l with I_l. Agent i learns about the rewards and *purchase rates* of its actions in each set in the partition independently from the other sets in the partition based on the context information of the users that arrived to agent i and the users for which agent i is recommended by another agent. Since users with similar contexts have similar purchase probabilities (Assumptions 1 and 2), it is expected that the optimal recommendations are similar for users located in the same set $I_l \in \mathcal{I}_T$. Since the best recommendations are learned independently for each set in $\mathcal{I}_T$, there is a tradeoff between the number of sets in $\mathcal{I}_T$ and the estimation of the best recommendations for contexts in each set in $\mathcal{I}_T$. We will show that in order to bound regret sublinearly over time, the parameter m_T should be non-decreasing in T.

There are two versions of CBMR: one for group-dependent purchase probabilities given by Assumption 1, which is called CBMR-d, and the other for independent purchase probabilities given by Assumption 2, which is called CBMR-ind. The difference between these two is that CBMR-d calculates the expected reward from each action in $\mathcal{L}_i$ for agent i separately, while CBMR-ind forms the expected reward of each action in $\mathcal{L}_i$ based on the expected rewards of the items recommended in the chosen action. We have

$$|\mathcal{L}_i| = \binom{|\mathcal{F}_i|}{N} + \sum_{n=0}^{N-1} \binom{|\mathcal{F}_i|}{n} \binom{N-n+M-2}{M-1},$$

Train(k, n, t):
Select action $\alpha_i(t) = k$.
Receive reward $\tilde{r}_k(t) := O^i_{\boldsymbol{\alpha},\boldsymbol{\beta}}(t, x_i(t))$ given in (4).
$n++$.

Explore(k, n, r, t):
Select action $\alpha_i(t) = k$.
Receive reward $\tilde{r}_k(t) := O^i_{\boldsymbol{\alpha},\boldsymbol{\beta}}(t, x_i(t))$ given in (4).
$r = (nr + \tilde{r}_k(t))/(n+1)$.
$n++$.

Exploit($\boldsymbol{n}$, $\boldsymbol{r}$, t):
Select action $\alpha_i(t) \in \arg\max_{k \in \mathcal{L}_i} r_k$.
Receive reward $\tilde{r}_{\alpha_i(t)}(t) := O^i_{\boldsymbol{\alpha},\boldsymbol{\beta}}(t, x_i(t))$ given in (4).
$r_{\alpha_i(t)} = (n_{\alpha_i(t)} r_{\alpha_i(t)} + \tilde{r}_{\alpha_i(t)}(t))/(n_{\alpha_i(t)} + 1)$.
$n_{\alpha_i(t)} ++$.

Explore2(k, n, π, j, t):
Recommend set of items $\mathcal{F}_i(k)$ to agent j.
Receive purchase feedback $\tilde{\pi}_k(t) := U^i_{\boldsymbol{\alpha},\boldsymbol{\beta}}(t, x_j(t))$ given in (5) and the commission.
$\pi = (n\pi + \tilde{\pi}_k(t))/(n+1)$.
$n++$.

Exploit2($\boldsymbol{n}$, $\boldsymbol{\pi}$, $\boldsymbol{m}$, j, t):
Select action $\beta_i(t) \in \arg\max_{k \in \mathcal{B}_i(\boldsymbol{m})} \pi_k$.
Recommend set of items $\mathcal{F}_i(\beta_i(t))$ to agent j.
Receive purchase feedback $\tilde{\pi}_{\beta_i(t)}(t) := U^i_{\boldsymbol{\alpha},\boldsymbol{\beta}}(t, x_j(t))$ given in (5) and the commission.
$\pi_{\beta_i(t)} = (n_{\beta_i(t)} \pi_{\beta_i(t)} + \tilde{\pi}_{\beta_i(t)}(t))/(n_{\beta_i(t)} + 1)$.
$n_{\beta_i(t)} ++$.

Fig. 3. Pseudocode of the training, exploration and exploitation modules for agent i.

which grows combinatorially in N and M and polynomially in $|\mathcal{F}_i|$. We will show that CBMR-ind provides much faster learning than CBMR-d when Assumption 2 holds. We explain these algorithms in the following subsections.

A. Definition of CBMR-d

The pseudocode of CBMR-d is given in Fig. 2. At any time t in which agent i chooses an action $k_i \in \mathcal{L}_i$, for any agent $j \in \mathcal{M}_i(k_i)$, it should send a request to that agent for $m_j(k_i)$ recommendations along with the context of its user $x_i(t)$ and the recommendation vector $\boldsymbol{m}(k_i)$. If the agents do not want to share the set of recommendations and prices they made for agent i's user with agent i, they can use the regulator scheme proposed in Section III-C. CBMR-d can be in any of the following three phases at any time slot t: the *training* phase in which agent i chooses an action k_i such that it trains another agent j by asking it for recommendations and providing i's user's context information x, so that agent j will learn to recommend its set of $m_j(k_i)$ items with the highest *purchase rate* for a user with context x, the *exploration* phase in which agent i forms accurate estimates of the expected reward of actions $k_i \in \mathcal{L}_i$ by selecting k_i and observing the purchases, and the *exploitation* phase in which agent i selects the action in $\mathcal{L}_i$ with the highest estimated reward to maximize its total reward. The pseudocodes of these phases are given in Fig. 3.

At each time t, agent i first checks which set in the partition $\mathcal{I}_T$ context $x_i(t)$ belongs to. We separate the set of actions in $\mathcal{L}_i$ into two. Let

$$\hat{\mathcal{L}}_i = \{(\boldsymbol{u}_i, \boldsymbol{n}_i) \in \mathcal{L}_i \text{ such that } n_{u_i} = 0,\ \forall u_i \in \mathcal{M}_{-i}\},$$

be the set of actions in which all recommendations are from $\mathcal{F}_i$, and $\tilde{\mathcal{L}}_i = \mathcal{L}_i - \hat{\mathcal{L}}_i$ be the set of actions in which at least one recommendation is from an agent in $\mathcal{M}_{-i}$.

The training phase is required for actions in $\tilde{\mathcal{L}}_i$, while only exploration and exploitation phases are required for actions in $\hat{\mathcal{L}}_i$. When agent i chooses an action $k_i \in \tilde{\mathcal{L}}_i$, the agents $j \in \mathcal{M}_i(k_i)$ recommend $m_j(k_i)$ items from $\mathcal{F}_j$ to agent i. Recall that $\mathcal{N}_i(t)$ denotes the set of N items that is recommended to agent i's user at time t, based on the actions chosen by agent i and the recommendations chosen by other agents for agent i. For $k_i \in \mathcal{L}_i$, let $N^i_{k_i,l}(t)$ be the number of times action k_i is selected in response to a context arriving to the set $I_l \in \mathcal{I}_T$ by time t. Since agent i does not know $\mathcal{F}_j$ and the purchase rates for $j \in \mathcal{M}_i(k_i)$, before forming estimates about the reward of action k_i, it needs to make sure that j will almost always recommend its set of $m_j(k_i)$ items with the highest purchase rate for agent i's user. Otherwise, agent i's estimate about the reward of action k_i might deviate a lot from the correct reward of action k_i with a high probability, resulting in agent i choosing suboptimal actions most of the time, hence resulting in a high regret. This is why the training phase is needed for actions in $\tilde{\mathcal{L}}_i$.

In order to separate training, exploration and exploitation phases, agent i keeps two counters for actions $k_i \in \tilde{\mathcal{L}}_i$ for each $I_l \in \mathcal{I}_T$. The first one, i.e., $N^i_{1,k_i,l}(t)$, counts the number of user arrivals to agent i with context in set I_l in the training phases of agent i by time t, in which it had chosen action k_i. The second one, i.e., $N^i_{2,k_i,l}(t)$, counts the number of user arrivals to agent i with context in set I_l in the exploration and the exploitation phases of agent i by time t, in which it had chosen action k_i. The observations made at the exploration and the exploitation phases are used to estimate the expected reward of agent i from taking action k_i. In addition to the counters $N^i_{k_i,l}(t)$, $N^i_{1,k_i,l}(t)$ and $N^i_{2,k_i,l}(t)$, agent i also keeps *control functions*, $D_1(t)$, $D_{2,k_i}(t)$, $k_i \in \tilde{\mathcal{L}}_i$ and $D_3(t)$, which are used together with the counters determine when to train, explore or exploit. The control functions are deterministic and non-decreasing functions of t that are related to the minimum number of observations of an action that is required so that the estimated reward of that action is sufficiently accurate to get a low regret in exploitations. We will specify the exact values of these functions later when we prove our regret results.

In addition to the counters and control functions mentioned above, for each $k_i \in \mathcal{L}_i$ and $I_l \in \mathcal{I}_T$, agent i keeps a *sample mean reward* $\bar{r}^i_{k_i,l}(t)$ which is the sample mean of the rewards (sum of prices of sold items and commissions divided by number of times k_i is selected except the training phase) agent i obtained from its recommendations when it selected action k_i in its exploration and exploitation phases for its own user with context in I_l by time t. Agent i also keeps a *sample mean purchase rate* $\bar{\pi}^i_{k_i,l}(t)$ which is the sample mean of the number of agent i's own items in $\mathcal{F}_i(k_i)$ sold to a user (either agent i's own user or user of another agent) with context in set I_l when agent i selects action k_i for that user.[5] Note that when agent i chooses action k_i, agent $j \in \mathcal{M}_{-i}$ has $\binom{|\mathcal{F}_j|}{m_j(k_i)}$ different sets of items to recommend to agent i. In order for agent j to recommend to agent i its best set of items that maximizes the commission agent i will get, agent j must have accurate *sample mean purchase rates* $\bar{\pi}^j_{k_j,l}(t)$ for its own actions $k_j \in \mathcal{B}_j(\boldsymbol{m}(k_i))$. Therefore, when a user with context $x_i(t) \in I_l$ arrives at time t, agent i checks if the following set is nonempty:

$$\mathcal{S}_{i,l}(t) := \Big\{ k_i \in \hat{\mathcal{L}}_i \text{ such that } N^i_{k_i,l}(t) \le D_1(t) \text{ or } k_i \in \tilde{\mathcal{L}}_i \text{ such that } N^i_{1,k_i,l}(t) \le D_{2,k_i}(t) \text{ or } N^i_{2,k_i,l}(t) \le D_3(t) \Big\},$$

in order to make sure that it has accurate estimates of the expected rewards of each of its actions $k_i \in \mathcal{L}_i$, as well as to make sure that other agents have accurate estimates about the *purchase rates* of their own set of items for a user with context in set I_l.

For $k_i \in \tilde{\mathcal{L}}_i$, let $\mathcal{E}^i_{k_i,l}(t)$ be the set of rewards collected from selections of action k_i at times $t' \in [t]$ when agent i's user's context is in set I_l, and all the other agents in $\mathcal{M}_i(k_i)$ are trained sufficiently, i.e., $N^i_{1,k_i,l}(t') > D_{2,k_i}(t')$. For $k_i \in \hat{\mathcal{L}}_i$, let $\mathcal{E}^i_{k_i,l}(t)$ be the set of rewards collected from action k_i by time t. If $\mathcal{S}_{i,l}(t) \neq \emptyset$, then agent i trains or explores by randomly choosing an action $\alpha_i(t) \in \mathcal{S}_{i,l}(t)$. If $\mathcal{S}_{i,l}(t) = \emptyset$, this implies that all actions in $\mathcal{L}_i$ have been trained and explored sufficiently, so that agent i exploits by choosing the action with the highest sample mean reward, i.e.,

$$\alpha_i(t) \in \arg\max_{k_i \in \mathcal{L}_i} \bar{r}^i_{k_i,l}(t). \tag{7}$$

We have $\bar{r}^i_{k_i,l}(t) = \left(\sum_{r \in \mathcal{E}^i_{k_i,l}(t)} r\right)/|\mathcal{E}^i_{k_i,l}(t)|$.[6] When there is more than one action which has the highest sample mean reward, one of them is randomly selected.

The other part of CBMR-d, i.e., β_i, gives agent i the set of items to recommend to agent j when agent j takes an action $\alpha_j(t) = k_j \in \mathcal{L}_j$ for which $i \in \mathcal{M}_j(k_j)$, and sends to agent i its user's context $x_j(t) \in I_l \in \mathcal{I}_T$ and the recommendation vector $\boldsymbol{m}(k_j)$. In order to recommend the set of items with the maximum *purchase rate* for the user of agent j, agent i should learn the *purchase rate* of its own items for the recommendation vector $\boldsymbol{m}(k_j)$. Agent i responds to agent j's request in the following way. If there is any $k_i \in \mathcal{B}(\boldsymbol{m}(k_j))$ for which the *purchase rate* is under-explored, i.e., $N^i_{k_i,l}(t) \le D_1(t)$, then agent i recommends to agent j the set of items $\mathcal{F}_i(k_i)$. Otherwise if all *purchase rates* of actions in $\mathcal{B}(\boldsymbol{m}(k_j))$ are explored sufficiently, i.e., $N^i_{k_i,l}(t) > D_1(t)$ for all $k_i \in \mathcal{B}(\boldsymbol{m}(k_j))$, then agent i recommends to agent j the set of its own items which maximizes the *sample mean purchase rate*, i.e., $\mathcal{F}_i(\hat{k}^*_i(k_j,l,t))$, where

$$\hat{k}^*_i(k_j,l,t) = \arg\max_{k_i \in \mathcal{B}(\boldsymbol{m}(k_j))} \bar{\pi}^i_{k_i,l}(t).$$

In the following subsection we prove an upper bound on the regret of CBMR-d.

B. Analysis of the Regret of CBMR-d

For each $I_l \in \mathcal{I}_T$ and $k_i \in \mathcal{L}_i$, let $\overline{\mu}_{i,k_i,l} := \sup_{x \in I_l} \mu_{i,k_i}(x)$, $\underline{\mu}_{i,k_i,l} := \inf_{x \in I_l} \mu_{i,k_i}(x)$, $\overline{\pi}_{i,k_i,l} := \sup_{x \in I_l} \pi_{i,k_i}(x)$ and $\underline{\pi}_{i,k_i,l} := \inf_{x \in I_l} \pi_{i,k_i}(x)$. Let x^*_l be the context at the center

[5] We will explain how agent i selects an action k_i when agent j requests items from i when we describe $\beta_i^{\text{CMBR-}d}$.

[6] Agent i does not need to keep $\mathcal{E}^i_{k_i,l}(t)$ in its memory. Keeping and updating $\bar{r}^i_{k_i,l}(t)$ online, as new rewards are observed is enough.

of the set I_l. We define the *optimal reward* action of agent i for set I_l as

$$k_i^*(l) := \arg\max_{k_i \in \mathcal{L}_i} \mu_{i,k_i}(x_l^*),$$

and the *optimal purchase rate* action of agent j for agent i given agent i selects action k_i for set I_l as

$$k_j^*(k_i, l) := \arg\max_{k_j \in \mathcal{B}_j(\boldsymbol{m}(k_i))} \pi_{j,k_j}(x_l^*).$$

Let

$$\mathcal{K}^i_{\theta,a_1,l}(t) := \left\{ k_i \in \mathcal{L}_i : \underline{\mu}_{i,k_i^*(l),l} - \overline{\mu}_{i,k_i,l} > a_1 t^\theta \right\},$$

be the set of *suboptimal reward* actions for agent i at time t, and

$$\mathcal{Y}^{j,k_i}_{\theta,a_1,l}(t) := \left\{ k_j \in \mathcal{B}_j(\boldsymbol{m}(k_i)) : \underline{\pi}_{i,k_j^*(k_i,l),l} - \overline{\pi}_{i,k_j,l} > a_1 t^\theta \right\},$$

be the set of *suboptimal purchase rate* actions of agent j for agent i, given agent i chooses action k_i at time t, where $\theta < 0$, $a_1 > 0$. The agents are not required to know the values of the parameters θ and a_1. They are only used in our analysis of the regret. First, we will give regret bounds that depend on values of θ and a_1, and then we will optimize over these values to find the best bound. Let $Y_R := \sup_{x \in \mathcal{X}} \mu_{i,k_i^*(x)}(x)$, which is the maximum expected loss agent i can incur for a time slot in which it chooses a suboptimal action.

The regret given in (6) can be written as a sum of three components:

$$R^i(T) = E[R^i_e(T)] + E[R^i_s(T)] + E[R^i_n(T)],$$

where $R^i_e(T)$ is the regret due to training and explorations by time T, $R^i_s(T)$ is the regret due to suboptimal action selections in exploitations by time T, and $R^i_n(T)$ is the regret due to near optimal action ($k_i \in \mathcal{L}_i - \mathcal{K}^i_{\theta,a_1,l}(t)$) selections in exploitations by time T of agent i, which are all random variables. In the following lemmas we will bound each of these terms separately. The following lemma bounds $E[R^i_e(T)]$. Due to space limitations we give the lemmas in this and the following subsections without proofs. The proofs can be found in our online appendix [31].

Lemma 1: When CBMR-d is run with parameters $D_1(t) = D_3(t) = t^z \log t$, $D_{2,k_i}(t) = \max_{j \in \mathcal{M}_i(k_i)} \binom{F_{\max}}{m_j(k_i)} t^z \log t$, and $m_T = \lceil T^\gamma \rceil$,[7] where $0 < z < 1$ and $0 < \gamma < 1/d$, we have

$$E[R^i_e(T)] \le Y_R 2^d \left(|\mathcal{L}_i| + |\tilde{\mathcal{L}}_i| \binom{F_{\max}}{\lceil \frac{F_{\max}}{2} \rceil} \right) T^{z+\gamma d} \log T + Y_R 2^d |\mathcal{L}_i| T^{\gamma d}.$$

From Lemma 1, we see that the regret due to explorations is linear in the number of sets in partition $\mathcal{I}_T$, i.e., $(m_T)^d$, hence exponential in parameters γ and z. We conclude that z and γ should be small to achieve sublinear regret in training and exploration phases.

For any $k_i \in \mathcal{L}_i$ and $I_l \in \mathcal{I}_T$, the sample mean $\bar{r}^i_{k_i,l}(t)$ represents a random variable which is the average of the independent samples in set $\mathcal{E}^i_{k_i,l}(t)$. Different from classical finite-armed bandit theory [13], these samples are not identically distributed. In order to facilitate our analysis of the regret, we generate two different artificial i.i.d. processes to bound the probabilities related to $\bar{r}^i_{k_i,l}(t)$, $k_i \in \mathcal{L}_i$. The first one is the *best* process in which rewards are generated according to a bounded i.i.d. process with expected reward $\overline{\mu}_{i,k_i,l}$, the other one is the *worst* process in which rewards are generated according to a bounded i.i.d. process with expected reward $\underline{\mu}_{i,k_i,l}$. Let $r^{\text{best}}_{k_i,l}(\tau)$ denote the sample mean of the τ samples from the best process and $r^{\text{worst}}_{k_i,l}(\tau)$ denote the sample mean of the τ samples from the worst process. We will bound the terms $E[R^i_n(T)]$ and $E[R^i_s(T)]$ by using these artificial processes along with the Hölder condition given in Assumption 1. Details of this are given in [31]. The following lemma bounds $E[R^i_s(T)]$.

Lemma 2: When CBMR-d is run with parameters $D_1(t) = D_3(t) = t^z \log t$, $D_{2,k_i}(t) = \max_{j \in \mathcal{M}_i(k_i)} \binom{F_{\max}}{m_j(k_i)} t^z \log t$, and $m_T = \lceil T^\gamma \rceil$, where $0 < z < 1$ and $0 < \gamma < 1/d$, given that $2LY_R(\sqrt{d})^\alpha t^{-\gamma\alpha} + 2(Y_R+2)t^{-z/2} \le a_1 t^\theta$, we have

$$E[R^i_s(T)] \le 2^{d+1} Y_R |\hat{\mathcal{L}}_i| \beta_2 T^{\gamma d} + \frac{2^{d+2} Y_R |\tilde{\mathcal{L}}_i| \beta_2 T^{\gamma d + z/2}}{z}.$$

From Lemma 2, we see that the regret increases exponentially with parameters γ and z, similar to the result of Lemma 1. These two lemmas suggest that γ and z should be as small as possible, given that the condition $2LY_R(\sqrt{d})^\alpha t^{-\gamma\alpha} + 2(Y_R+2)t^{-z/2} \le a_1 t^\theta$ is satisfied.

When agent i chooses an action $k_i \in \mathcal{L}_i$ such that $j \in \mathcal{M}_i(k_i)$, there is a positive probability that agent j will choose a suboptimal set of $m_j(k_i)$ items to recommend to agent i's user, i.e., it will choose a *suboptimal purchase rate action* for agent i's action. Because if this, even if agent i chooses a near optimal action $k_i \in \mathcal{L}_i - \mathcal{K}^i_{\theta,a_1,l}(t)$ it can still get a low reward. We need to take this into account in order to bound $E[R^i_n(T)]$. The following lemma gives the bound on $E[R^i_n(T)]$.

Lemma 3: When CBMR-d is run with parameters $D_1(t) = D_3(t) = t^z \log t$, $D_{2,k_i}(t) = \max_{j \in \mathcal{M}_i(k_i)} \binom{F_{\max}}{m_j(k_i)} t^z \log t$, and $m_T = \lceil T^\gamma \rceil$, where $0 < z < 1$ and $0 < \gamma < 1/d$, given that $2LY_R(\sqrt{d})^\alpha t^{-\gamma\alpha} + 2(Y_R+2)t^{-z/2} \le a_1 t^\theta$, we have

$$E[R^i_n(T)] \le \frac{(2a_1 T^{1+\theta})}{(1+\theta)} + 2Y_R \binom{F_{\max}}{\lceil \frac{F_{\max}}{2} \rceil} \beta_2.$$

From Lemma 3, we see that the regret due to near optimal actions depends exponentially on θ which is related to the negative of γ and z. Therefore, γ and z should be chosen as large as possible to minimize the regret due to near optimal actions. Combining the above lemmas, we obtain the finite time regret bound for agents using CBMR-d, which is given in the following theorem.

Theorem 1: When CBMR-d is run with parameters $D_1(t) = D_3(t) = t^{2\alpha/(3\alpha+d)} \log t$, $D_{2,k_i}(t) = \max_{j \in \mathcal{M}_i(k_i)} \binom{F_{\max}}{m_j(k_i)} t^{2\alpha/(3\alpha+d)} \log t$, and $m_T = \lceil T^{1/(3\alpha+d)} \rceil$, we have

$$\begin{aligned} R^i(T) &\le T^{(2\alpha+d)/(3\alpha+d)} \left(\frac{4(Y_R(Ld^{\alpha/2}+1)+1)}{\frac{(2\alpha+d)}{(3\alpha+d)}} + Y_R 2^d Z_i \log T \right) \\ &\quad + T^{(\alpha+d)/(3\alpha+d)} \frac{2^{d+2} Y_R |\tilde{\mathcal{L}}_i| \beta_2}{\frac{2\alpha}{(3\alpha+d)}} \\ &\quad + T^{d/(3\alpha+d)} 2^d Y_R (2|\tilde{\mathcal{L}}_i|\beta_2 + |\mathcal{L}_i|) + 2Y_R \binom{F_{\max}}{\lceil \frac{F_{\max}}{2} \rceil} \beta_2, \end{aligned}$$

i.e., $R_i(T) = O\left(Z_i T^{(2\alpha+d)/(3\alpha+d)}\right)$, where $Z_i = |\mathcal{L}_i| + |\tilde{\mathcal{L}}_i| \binom{F_{\max}}{\lceil F_{\max}/2 \rceil}$.

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

Proof: The highest orders of regret come from explorations and near optimal arms, which are $O(T^{\gamma d+z})$ and $O(T^{1+\theta})$ respectively. We need to optimize them with respect to the constraint $2LY_R(\sqrt{d})^\alpha t^{-\gamma\alpha} + 2(Y_R+2)t^{-z/2} \leq a_1 t^\theta$, which is assumed in Lemmas 2 and 3. The values that minimize the regret for which this constraint holds are $\theta = -z/2, \gamma = z/(2\alpha)$ $a_1 = 2Y_R(Ld^{\alpha/2}+1)+4$ and $z = 2\alpha/(3\alpha+d)$. The result follows from summing the bounds in Lemmas 1, 2 and 3. ■

Remark 2: A uniform partition of the context space such as $\mathcal{I}_T$ may not be very efficient when user arrivals have contexts that are concentrated into some (unknown) regions of the context space. In such a case, it is better to explore and train these regions more frequently than regions with few context arrivals. Algorithms that start with the entire context space as a single set and then adaptively partition the context space into smaller regions as more and more users arrive may achieve faster convergence rates, i.e., smaller regret, for the types of arrivals mentioned above. Such an algorithm that will work for $N = 1$ is given in [32].

Theorem 1 indicates that agent i can achieve sublinear regret with respect to the best "oracle" recommendation strategy which knows the purchase probabilities of all the items in $\mathcal{F}$. However, the learning rate of CBMR-d can be slow when M, N and $\mathcal{F}$ are large since the set of actions is combinatorial in these parameters. As a final remark, since regret is sublinear, the average reward of CBMR-d converges to the average reward of the best "oracle" recommendation strategy, i.e., $\lim_{T\to\infty} R^i(T)/T = 0$.

C. Definition of CBMR-ind

In this subsection we describe the learning algorithm CBMR-ind. We assume that Assumption 2 holds. Let $\mathcal{J}_{i,j} := \{1_j, 2_j, \ldots, N_j\}$ denote the set of the number of recommendations agent i can request from agent j, where we use the subscript j to denote that the recommendations are requested from agent j. Let $\tilde{\mathcal{J}}_i := \cup_{j\in\mathcal{M}_{-i}}\mathcal{J}_{i,j}$, and $\mathcal{J}_i := \mathcal{F}_i \cup \tilde{\mathcal{J}}_i$ be the set of *arms* of agent i. We have $|\mathcal{J}_i| = |\mathcal{F}_i| + (M-1)N$. We denote an arm of agent i by index u. For arm $u \in \tilde{\mathcal{J}}_i$, let $j(u)$ be the agent that is called for item recommendations to agent i's user, and let $n(u)$ be the number of requested items from agent $j(u)$.[8] For $u \in \mathcal{F}_i$, $j(u) = i$ and $n(u) = 1$. In CBMR-ind, at each time t, agent i chooses a set of arms such that the total number of item recommendations it makes to its user is N. It is evident that choosing such a set of arms is equivalent to choosing an action $k_i \in \mathcal{L}_i$. Every action $k_i \in \mathcal{L}_i$ maps to a unique set of arms. Thus, for an action $k_i \in \mathcal{L}_i$, let $\mathcal{A}(k_i)$ be the set of arms corresponding to k_i.

Let $\mathcal{G}_{j,n}(x) \subset \mathcal{F}_j$ be the set of n items in $\mathcal{F}_j$ with the highest purchase probabilities for a user with context x. For an arm $u \in \mathcal{F}_i$, let its *purchase rate* for a user with context x be $\nu_{i,u}(x) := q_u(x)$, and for an arm $u \in \tilde{\mathcal{J}}_i$, let it be $\nu_{i,u}(x) := \sum_{f\in\mathcal{G}_{j(u),n(u)}} q_f(x)$. Since Assumption 2 holds, we have

$$\mu_{i,k_i}(x) = \sum_{u\in\mathcal{F}_i\cap\mathcal{A}(k_i)} p_u^i \nu_{i,u}(x) + \sum_{u\in\mathcal{A}(k_i)-\mathcal{F}_i} c_{i,j(u)}\nu_{i,u}(x).$$

[8] $j(u)$, $u \in \tilde{\mathcal{J}}_i$ is different from $j(f)$, $f \in \mathcal{F}$, which denotes the agent that owns item f, and $n(u)$ is different from $n_j(k_i)$, which denotes the number of items agent j should recommend when agent i chooses action $k_i \in \mathcal{L}_i$.

Context Based Multiple Recommendations for independent purchase probabilities (CBMR-ind for agent i):

Input: $D_1(t)$, $D_{2,u}(t)$, $u \in \tilde{\mathcal{J}}_i$, $D_3(t)$, T, m_T.
Initialize: Partition $[0,1]^d$ into $(m_T)^d$ sets, indexed by the set $\mathcal{I}_T = \{1, 2, \ldots, (m_T)^d\}$.
$N^i_{u,l} = 0, \forall u \in \mathcal{J}_i, I_l \in \mathcal{I}_T$,
$N^i_{1,u,l} = 0, N^i_{2,u,l} = 0, \forall u \in \tilde{\mathcal{J}}_i, l \in \mathcal{I}_T$.
$\bar{\nu}^i_{u,l} = 0, \forall u \in \mathcal{J}_i, I_l \in \mathcal{I}_T$.
while $t \geq 1$ **do**
  for $l = 1, \ldots, (m_T)^d$ **do**
    if $x_i(t) \in I_l$ **then**
      if $\mathcal{S}^{\text{ind}}_{i,l}(t) \neq \emptyset$ (given in (8)) **then**
        Choose $\alpha_i(t)$ randomly from the subset of $\mathcal{L}_i$ such that $\mathcal{A}(\alpha_i(t)) \cap \mathcal{S}^{\text{ind}}_{i,l}(t) \neq \emptyset$.
      else
        Choose $\alpha_i(t)$ as in (9).
      end if
    end if
  end for
  Receive reward $O^i_{\boldsymbol{\alpha},\boldsymbol{\beta}}(t, x_i(t))$ given in (4).
  for $u \in \mathcal{A}(\alpha_i(t))$ **do**
    if $u \in \tilde{\mathcal{J}}_i$ and $N^i_{1,u,l} \leq D_{2,u}(t)$ **then**
      $N^i_{1,u,l} + +$.
    else if $u \in \tilde{\mathcal{J}}_i$ and $N^i_{1,u,l} > D_{2,u}(t)$ **then**
      $\bar{\nu}^i_{u,l} = \frac{N^i_{2,u,l}\bar{\nu}^i_{u,l} + U^{j(u)}_{\boldsymbol{\alpha},\boldsymbol{\beta}}(t, x_i(t))}{N^i_{2,u,l}+1}$ (given in (5)).
      $N^i_{2,u,l} + +$.
    else
      $\bar{\nu}^i_{u,l} = (N^i_{u,l}\bar{\nu}^i_{u,l} + Y^{\alpha_i}_u(t))/(N^i_{2,u,l}+1)$.
      $N^i_{u,l} + +$.
    end if
  end for
  $t = t + 1$.
end while

Fig. 4. Pseudocode of CBMR-ind for agent i.

Different from CBMR-d, which estimates the reward of each action in $\mathcal{L}_i$ separately, CBMR-ind estimates the *purchase rates* of the arms in $\mathcal{J}_i$, and uses them to construct the estimated rewards of actions in $\mathcal{L}_i$. The advantage of CBMR-ind is that the purchase rate estimate of an arm $u \in \mathcal{J}_i$ can be updated based on the purchase feedback whenever any action $\mathcal{L}_i$ that contains arm u is selected by agent i.

The pseudocode of CBMR-ind is given in Fig. 4. CMBR-ind partitions the context space in the same way as CBMR-d. Unlike CBMR-d, CBMR-ind does not have exploration, exploitation and training phases for actions $k_i \in \mathcal{L}_i$. Rather than that, it has exploration and exploitation phases for each arm $u \in \mathcal{F}_i$, and exploration, exploitation and training phases for each arm $u \in \tilde{\mathcal{J}}_i$. Since a combination of arms is selected at each time slot, selected arms can be in different phases. For each $u \in \mathcal{J}_i$ and $I_l \in \mathcal{I}_T$, CBMR-ind keeps the *sample mean purchase rate* $\bar{\nu}^i_{u,l}(t)$, which is the sample mean of the number of purchased items corresponding to arm u, purchased by users with contexts in I_l in exploration and exploitation phases of agent i by time t.

Similar to the counters and control functions of CBMR-d, CBMR-ind also keeps counters and control functions for each $u \in \mathcal{J}_i$ and $I_l \in \mathcal{I}_T$. Basically, for $u \in \mathcal{J}_i$, $N^i_{u,l}(t)$ counts the number of times arm u is selected by agent i to make a recommendation to a user with context in I_l by time t. Counters $N^i_{1,u,l}(t)$ and $N^i_{2,u,l}(t)$ are only kept for arms $u \in \tilde{\mathcal{J}}_i$. The former one counts the number of times arm u is trained by agent i for times its users had contexts in I_l by time t, while the latter

one counts the number of times arm u is explored and exploited by agent i for times its users had contexts in I_l by time t. Let

$$\mathcal{S}_{i,l}^{\text{ind}}(t) := \Big\{ u \in \mathcal{F}_i \text{ such that } N_{u,l}^i(t) \le D_1(t) \text{ or } u \in \tilde{\mathcal{J}}_i \text{ such that } N_{1,u,l}^i(t) \le D_{2,u}(t) \text{ or } N_{2,u,l}^i(t) \le D_3(t) \Big\}, \quad (8)$$

where $D_1(t)$, $D_{2,u}(t)$, $u \in \tilde{\mathcal{J}}_i$ and $D_3(t)$ are counters similar to the counters of CBMR-d. Assume that $x_i(t) \in I_l$. If $\mathcal{S}_{i,l}^{\text{ind}}(t) \neq \emptyset$, then agent i randomly chooses an action $\alpha_i(t) \in \mathcal{L}_i$ from the set of actions for which $\mathcal{A}(k_i) \cap \mathcal{S}_{i,l}^{\text{ind}}(t) \neq \emptyset$. Else if $\mathcal{S}_{i,l}^{\text{ind}}(t) = \emptyset$, then agent i will choose the action

$$\alpha_i(t) = \arg\max_{k_i \in \mathcal{L}_i} \sum_{u \in \mathcal{F}_i \cap \mathcal{A}(k_i)} p_u^i \bar{\nu}_{u,l}^i(t) + \sum_{u \in \mathcal{A}(k_i) - \mathcal{F}_i} c_{i,j(u)} \bar{\nu}_{u,l}^i(t). \quad (9)$$

After action $\alpha_i(t)$ is chosen, the counters and sample mean purchase rates of arms in $\mathcal{A}(\alpha_i(t))$ are updated based on in which phase they are in.

Agent i responds to item requests of other agents with users with contexts in $I_l \in \mathcal{I}_T$ in the following way. Any under-explored item $u \in \mathcal{F}_i$, i.e., $N_{u,l}^i(t) \le D_1(t)$ is given priority to be recommended. If the number of under-explored items is less than the number of requested items, then the remaining items are selected from the set of items in $u \in \mathcal{F}_i$ with the highest sample mean purchase rates such that $N_{u,l}^i(t) > D_1(t)$. The pseudocode of this is not given due to limited space.

D. Analysis of the Regret of CBMR-ind

In this subsection we bound the regret of CBMR-ind. Under Assumption 2, the expected reward of an item f to agent i for a user with context x is $\kappa_{i,f}(x) := p_f^i q_f(x)$ for $f \in \mathcal{F}_i$ and $\kappa_{i,f}(x) := c_{i,j} q_f(x)$ for $f \in \mathcal{F}_j$. For a set of items $\mathcal{N}$, let $f_n(\mathcal{N})$ denote the item in $\mathcal{N}$ with the nth highest expected reward for agent i. For an item $f \in \mathcal{F}$ and $I_l \in \mathcal{I}_T$, let $\underline{\kappa}_{i,f,l} := \inf_{x \in I_l} \kappa_{i,f}(x)$, and $\bar{\kappa}_{i,f,l} := \sup_{x \in I_l} \kappa_{i,f}(x)$. For the set I_l of the partition $\mathcal{I}_T$, the set of suboptimal arms of agent i at time t is given by

$$\mathcal{U}_{\theta,a_1,l}^i(t) := \Big\{ u \in \mathcal{F}_i : \underline{\kappa}_{i,f_N(\mathcal{F}),l} - \bar{\kappa}_{i,u,l} \ge a_1 t^\theta \Big\} \cup \Big\{ u \in \tilde{\mathcal{J}}_i : \underline{\kappa}_{i,f_N(\mathcal{F}),l} - \bar{\kappa}_{i,f_{n(u)}(\mathcal{F}_{j(u)}),l} \ge a_1 t^\theta \Big\}.$$

We will optimize over a_1 and θ as we did in Section IV-B. The set of near-optimal arms at time t for I_l is $\mathcal{J}_i - \mathcal{U}_{\theta,a_1,l}^i(t)$. Similar to the approach we took in Section IV-B, we divide the regret into three parts: $R_e^i(T)$, $R_s^i(T)$ and $R_n^i(T)$, and bound them individually. Different from the analysis of CBMR-d, here $R_s^i(T)$ denotes the regret in time slots in which all selected arms are exploited and at least one them is suboptimal, while $R_n^i(T)$ denotes the regret in time slots in which all selected arms are exploited and all of them are near-optimal. In the following lemma, we bound the regret of CBMR-ind due to explorations and trainings.

Lemma 4: When CBMR-ind is run by the agents with $D_1(t) = t^z \log t$, $D_{2,u}(t) = \binom{F_{\max}}{n(u)} t^z \log t$, $D_3(t) = t^z \log t$ and $m_T = \lceil T^\gamma \rceil$, where $0 < z < 1$ and $0 < \gamma < 1/d$, we have

$$E[R_e^i(T)] \le Y_R 2^d (|\mathcal{J}_i| + (M-1)N) T^{\gamma d} + Y_R 2^d \left(|\mathcal{J}_i| + (M-1) \sum_{z=1}^{N} \binom{F_{\max}}{z} \right) T^{z+\gamma d} \log T.$$

In the next lemma, we bound $E[R_s^i(T)]$.

Lemma 5: When CBMR-ind is run by the agents with $D_1(t) = t^z \log t$, $D_{2,u}(t) = \binom{F_{\max}}{n(u)} t^z \log t$, $D_3(t) = t^z \log t$ and $m_T = \lceil T^\gamma \rceil$, where $0 < z < 1$ and $0 < \gamma < 1/d$, given that $2LY_R(\sqrt{d})^\alpha t^{-\gamma\alpha} + 2(Y_R+2)t^{-z/2} \le a_1 t^\theta$, we have

$$E[R_s^i(T)] \le 2^{d+1} Y_R |\mathcal{J}_i| \beta_2 T^{\gamma d} + \frac{2^{d+2} N Y_R |\tilde{\mathcal{J}}_i| \beta_2 T^{\gamma d + z/2}}{z}.$$

Different from Lemma 2, $E[R_s^i(T)]$ is linear in $|\mathcal{J}_i|$ instead of in $|\mathcal{L}_i|$. In the next lemma, we bound the $E[R_n^i(T)]$.

Lemma 6: When CBMR-ind is run by the agents with $D_1(t) = t^z \log t$, $D_{2,u}(t) = \binom{F_{\max}}{n(u)} t^z \log t$, $D_3(t) = t^z \log t$ and $m_T = \lceil T^\gamma \rceil$, where $0 < z < 1$ and $0 < \gamma < 1/d$, given that $2LY_R(\sqrt{d})^\alpha t^{-\gamma\alpha} + 2(Y_R+2)t^{-z/2} \le a_1 t^\theta$, we have

$$E[R_n^i(T)] \le \frac{(2N a_1 T^{1+\theta})}{(1+\theta)} + 2Y_R N F_{\max} \beta_2.$$

Combining the above lemmas, we obtain the regret bound for agents using CBMR-ind.

Theorem 2: When CBMR-ind is run by the agents with $D_1(t) = t^{2\alpha/(3\alpha+d)} \log t$, $D_{2,u}(t) = \binom{F_{\max}}{n(u)} t^{2\alpha/(3\alpha+d)} \log t$, $D_3(t) = t^{2\alpha/(3\alpha+d)} \log t$ and $m_T = \lceil T^{1/(3\alpha+d)} \rceil$, we have

$$\begin{aligned} R^i(T) \le\ & T^{(2\alpha+d)/(3\alpha+d)} \left(\frac{4N(Y_R(Ld^{\alpha/2}+1)+1)}{\frac{(2\alpha+d)}{(3\alpha+d)}} + Y_R 2^d Z_i' \log T \right) \\ & + T^{(\alpha+d)/(3\alpha+d)} \frac{2^{d+2} N Y_R |\tilde{\mathcal{J}}_i| \beta_2}{\frac{2\alpha}{(3\alpha+d)}} \\ & + T^{d/(3\alpha+d)} 2^d Y_R (2|\mathcal{J}_i|\beta_2 + |\mathcal{J}_i| + (M-1)N) \\ & + 2Y_R N F_{\max} \beta_2, \end{aligned}$$

i.e., $R^i(T) = O\left(Z_i' T^{(2\alpha+d)/(3\alpha+d)}\right)$, where $Z_i' = |\mathcal{J}_i| + (M-1)\sum_{z=1}^{N} \binom{F_{\max}}{z}$.

Proof: The highest orders of regret come from $E[R_e^i(T)]$ and $E[R_n^i(T)]$, which are $O(T^{\gamma d+z})$ and $O(T^{1+\theta})$, respectively. We need to optimize them with respect to the constraint $2LY_R(\sqrt{d})^\alpha t^{-\gamma\alpha} + 2(Y_R+2)t^{-z/2} \le a_1 t^\theta$, which is assumed in Lemmas 5 and 6. This gives us $\theta = -z/2$, $\gamma = z/(2\alpha)$ $a_1 = 2Y_R(Ld^{\alpha/2}+1) + 4$ and $z = 2\alpha/(3\alpha+d)$. The result follows from summing the bounds in Lemmas 4, 5 and 6. ■

The result of Theorem 2 indicates that the regret of CMBR-ind is sublinear in time and has the same time order as the regret of CBMR-d. However, the regret of CMBR-ind depends linearly on $|\mathcal{J}_i|$, which is much better than the linear dependence of the regret of CBMR-d on $|\mathcal{L}_i|$.

We would also like to note that the number of trainings and explorations, and hence the regret, can be significantly reduced when agent i knows $|\mathcal{F}_j|$ for all $j \in \mathcal{M}_{-i}$. In this case, agent i can use the control function $D_{2,u}(t) := \binom{|\mathcal{F}_{j(u)}|}{n(u)} t^{2\alpha/(3\alpha+d)} \log t$ to decide if it needs to train arm u.

E. Comparison of CBMR-d and CBMR-ind

In this subsection we compare CBMR-d and CBMR-ind in terms of their regret bounds, training and exploration rates, and memory requirements. Note that the regret bound of CBMR-d depends on the size of the action space $\mathcal{L}_i$, which grows combinatorially with M and N, and is an N degree polynomial of $|\mathcal{F}_i|$. In contrast the size of the arm space $\mathcal{J}_i$ is just $|\mathcal{F}_i| + (M-1)N$. This is due to the fact that CBMR-d explores and exploits each action without exploiting the correlations between different actions. When the purchase probabilities depend on the set of items offered together, in the worst case there may be no correlation between rewards of different actions, and therefore the best one can do is to form independent sample mean estimates for each action in $\mathcal{K}_i$. However, when the purchase probabilities are independent of the items offered together, since the expected reward of agent i from an action $k_i \in \mathcal{L}_i$ is the sum of the expected rewards of the individual arms chosen by agent i in that action, substantial improvement over the regret bound is possible due to smaller number of explorations and trainings.

Another advantage of CBMR-ind is that it requires a significantly smaller amount of memory than CBMR-d. CBMR-d needs to keep sample mean rewards and sample mean purchase rates for all actions and for all partitions of the context space, while CBMR-ind only needs to keep sample mean purchase rates for all arms and for all partitions of the context space.

V. Performance as a Function of the Network

In our analysis in the previous section we assumed that all agents are directly connected to each other. In reality some agents may not be connected to each other. For example, agents i and j can both be connected to agent j', but there may not exist a direct link between agent i and agent j. This can happen when, for example, a pair of companies has a trade agreement between each other, but there is no trade agreement between companies i and j. We assume that whenever a trade link exists between agents i and j it is bidirectional. In this case, even though i cannot ask j for items to recommend, i can ask j' and j' can ask j for an item. Then, if i sells j's item, agent i will get commission $c_{i,j'}$ from j', while agent j' will get commission $c_{j',j} + c_{i,j'}$ from agent j so that it recovers the payment it made to agent i from agent j. We call agent j' the *relay agent*. Let $\mathcal{M}_i^{\text{dir}}$ denote the set of agents that are directly connected to agent i.

If agent i only cooperates with the agents in $\mathcal{M}_i^{\text{dir}}$ and all cooperating agents use CBMR-d or CBMR-ind, then agent i can achieve the sublinear regret bounds given in Theorems 1 and 2, with respect to the optimal distributed recommendation policy involving only the agents in $\{i\} \cup \mathcal{M}_i^{\text{dir}}$. However, since agent i cannot exploit the advantage of the items of other agents which are in $\mathcal{M}_{-i} - \mathcal{M}_i^{\text{dir}}$, its regret can be linear with respect to the optimal distributed recommendation policy involving all the agents.

CBMR-d and CBMR-ind can be modified in the following way to account for agents distributed over the network. Let $\mathcal{M}_{i,j}(p)$ be the set of relay agents between agents i and j when they are connected through path p. Let $D_p := |\mathcal{M}_{i,j}(p)|$. Let $h_1(p)$ be the agent that is connected directly to agent i in path p, let $h_2(p)$ be the agent that is connected directly to agent $h_1(p)$, and so on. Then, the commission framework is modified such that when agent i sells agent j's item it gets a commission $c_{i,h_1(p)}$ from agent $h_1(p)$, agent $h_1(p)$ gets a commission $c_{h_1(p),h_2(p)} + c_{i,h_1(p)}$ from agent $h_2(p)$, and so on, such that $c_{h_{D_p}(p),j} \leq p_f^j$, where $f \in \mathcal{F}_j$ is the item recommended by agent j to agent i's user. Using this scheme, all agents benefit from the series of transactions described above, thus it is better for them to cooperate based on the rules of the commission scheme than to act on their own. We assume that agent j will not recommend an item to agent $h_{D_p}(p)$ if $c_{h_{D_p}(p),j} > p_f^j$ for all $f \in \mathcal{F}_j$. Assume a connected network of agents $G(\mathcal{M}, E)$ in which the maximum degree is D_k and the longest path has D_h hops, where E denotes the set of direct links between the agents. Assume that the commissions $c_{i,j} > 0$ are given between any agent i and j with direct links. We define agent i's regret $R^i_{G(\mathcal{M},E)}(T)$ to be the difference between the expected total reward of the optimal policy for which agent i has access to the set of all the items of all agents in the network $G(\mathcal{M}, E)$ (but if the item is in $\mathcal{F}_j$, it gets the commission from agent j', which is the agent that is directly connected to agent i in the lowest commission path from agent i to agent j) and the expected total reward of the learning algorithm used by agent i. The exploration and training control functions are run using $(D_k)^{D_h} F_{\max}$ instead of $F_{\max}$. This way agent i will help the relay agents learn the other agent's recommendations accurately such that sublinear regret bounds can be achieved. The following theorem gives a bound on the regret of agents when they use the modified version of CBMR-ind discussed above, which we call CBMR-ind-N. A similar bound can also be proven for the modified version of CMBR-d.

Theorem 3: When Assumption 2 holds, if CBMR-ind-N is run by all agents in $G(\mathcal{M}, E)$, with agent i running CBMR-ind-N with the set of arms $\mathcal{J}_i$ as described in Section IV-C, defined using $\mathcal{M}_i^{\text{dir}}$ and $\mathcal{F}_i$ instead of $\mathcal{M}$ and $\mathcal{F}_i$, control functions $D_1(t) = D_3(t) = t^{2\alpha/(3\alpha+d)} \log t$, $D_{2,u}(t) = \binom{(D_k)^{D_h} F_{\max}}{n(u)} t^{2\alpha/(3\alpha+d)} \log t$, and $m_T = \lceil T^{1/(3\alpha+d)} \rceil$; and if commissions are such that all agents $j \in \mathcal{M}_{-i}$ will get a positive reward when their items are sold by agent i,[9] we have,

$$
\begin{aligned}
&R^i_{G(\mathcal{M},E)}(T) \\
&\leq T^{(\alpha+d)/(3\alpha+d)} \frac{2^{d+2} N Y_R |\tilde{\mathcal{J}}_i| \beta_2}{\frac{2\alpha}{(3\alpha+d)}} \\
&\quad + 2 Y_R N (D_k)^{D_h} F_{\max} \beta_2 \\
&\quad + T^{(2\alpha+d)/(3\alpha+d)} \left(\frac{4N(Y_R(Ld^{\alpha/2}+1)+1)}{\frac{(2\alpha+d)}{(3\alpha+d)}} + Y_R 2^d Z_i' \log T \right) \\
&\quad + T^{d/(3\alpha+d)} 2^d Y_R (2|\mathcal{J}_i|\beta_2 + |\mathcal{J}_i| + (M-1)N),
\end{aligned}
$$

[9]If the commission an agent needs to pay to sell an item to the user of agent i is greater than the price of all the items of that agent, then that agent can be removed from the set of agents which agent i can cooperate with. Then, our results will hold for the remaining set of agents.

i.e., $R^i_{G(\mathcal{M},E)}(T) = O\left(Z'_i T^{(2\alpha+d)/(3\alpha+d)}\right)$, where $Z'_i := |\mathcal{J}_i| + (M-1)\sum_{z=1}^{N}\binom{(D_k)^{D_h}F_{\max}}{z}$.

Proof: The proof is similar to the proof of Theorem 2, but more explorations and trainings are required to ensure that all agents in all paths learn the rewards of their arms accurately for each set in $\mathcal{I}_T$, before exploiting them. ■

Theorem 3 indicates that the regret increases exponentially in D_h and is an $D_h N$th degree polynomial in D_k. While this makes CBMR-ind-N impractical in non-small world networks, CBMR-ind-N can achieve small regret in small world networks in which most agents are not directly connected to each other but all agents can be reached a with small number of hops. Because of the additional commission the relay agent gets, an item which j will recommend when it is directly connected to agent i may not be recommended by j when it is connected via agent j'. This results in sub-optimality compared to the case when all agents are connected. In Section VI we numerically compare the effect of the agents' commissions on the performance.

More refined versions of Theorem 3 can be derived if we focus on specific networks. One interesting network structure is when there is a monopolist agent which is directly connected to all of the other agents, while other agents are only directly connected to the monopolist agent. Corollary 1 gives the regret bound for agent i when it is the monopolist agent, and Corollary 2 gives the regret bound for agent i when it is not the monopolist agent.

Corollary 1: When agent i is the monopolist agent, if it runs CBMR-ind-N with $D_{2,u}(t) = \binom{F_{\max}}{n(u)} t^{2\alpha/(3\alpha+d)} \log t$ and everything else remaining the same as in Theorem 3, it will achieve $R^i_{G_1(\mathcal{M},E)}(T) = O\left((\mathcal{F}_i + (M-1)N) T^{(2\alpha+d)/(3\alpha+d)}\right)$.

Corollary 2: When agent i is a non-monopolist agent, if it runs CBMR-ind-N with $D_{2,u}(t) = \binom{M^2 F_{\max}}{n(u)} t^{2\alpha/(3\alpha+d)} \log t$ and everything else remaining the same as in Theorem 3, it will achieve $R^i_{G_2(\mathcal{M},E)}(T) = O\left((\mathcal{F}_i + N) M^{2N} T^{(2\alpha+d)/(3\alpha+d)}\right)$.

Corollaries 1 and 2 imply that the learning is much faster, and hence the regret is much smaller when an agent has direct connections with more agents. This is because agent i learns directly about the purchase probabilities of items in agent j's inventory when it is directly connected to it, while the learning is indirect through a relay agent otherwise.

VI. Numerical Results

A. Description of the Data Set

The Amazon product co-purchasing network data set includes product IDs, sales ranks of the products, and for each product the IDs of products that are frequently purchased with that product. This data is collected by crawling the Amazon website [33] and contains 410,236 products and 3,356,824 edges between products that are frequently co-purchased together. We simulate CBMR-ind and CBMR-d using the following distributed data sets adapted based on Amazon data. For a set of N_1 chosen products from the Amazon data set, we take these products and other F_1 products that are frequently co-purchased with the set of N_1 products as our set of items.

The set of products that are taken in the first step of the above procedure is denoted by $\mathcal{C}_h$. The set of all products, i.e., $\mathcal{F}$, contains these N_1 products and the set of products frequently co-purchased with them, which we denote by $\mathcal{C}_f$. We assume that each item has a unit price of 1, but have different purchase probabilities for different types of users. Since user information is not present in the data set, we generate it artificially by assuming that every incoming user searches for a specific item in $\mathcal{C}_h$. This search query (item) will then be the context information of the user, hence the context space is $\mathcal{C}_h$. Thus, we set $\mathcal{I}_T = \mathcal{C}_h$. Based on this, the agent that the user arrives to recommends N items to the user. The agent's goal is to maximize the total number of items sold to its users.

We generate the purchase probabilities in the following way: For group-dependent purchase probabilities, when n of the products recommended for context $x \in \mathcal{C}_h$ are in the set of frequently co-purchased products with item x, then the purchase probability of each of these products will be $g_c(n) = 1 - an$, where $0 < a < 1/N$. For the other $N - n$ products which are not frequently co-purchased with item x, their purchase probability is $g_{nc} = b$, where $0 < b < 1 - a$. For independent purchase probabilities, when a product recommended for context x is in the set of frequently co-purchased products with item x, the purchase probability of that product will be g_c. When it is not, the purchase probability of that product will be g_{nc}, for which we have $g_c > g_{nc}$.

We assume that there are 3 agents and evaluate the performance of agent 1 based on the number of users arriving to agent 1 with a specific context x^*, which we take as the first item in set $\mathcal{C}_h$. We assume that $T = 100,000$, which means that 100,000 users with context x^* arrive to agent 1. Since the arrival rate of context x^* can be different for the agents, we assume arrivals with context x^* to other agents are drawn from a random process. We take $N_1 = 20$, $F_1 = 2$ and $N = 2$. As a result, we get 30 distinct items in $\mathcal{F}$ which are distributed among the agents such that $|\mathcal{F}_i| = 10$ for every agent i. Since the context space is discrete we have $d = 1$, and there is no Hölder condition on the purchase probabilities as given in Assumptions 1 and 2, hence we take $\alpha = 1/13$ such that $2\alpha/(3\alpha + d) = 1/8$. Unless otherwise stated, we assume that $g_c = 0.1$, $g_{nc} = 0.01$, $a = 0.5$ and $c_{i,j} = 0.5$.

B. Comparison of the Reward and Regret of CBMR-d and CBMR-ind for Group-Dependent Purchase Probabilities

We run both CBMR-d and CBMR-ind for group-dependent purchase probabilities assuming that both items that are frequently co-purchased with context x^* are in agent 1's inventory. The "oracle" optimal policy recommends one of the frequently co-purchased items and another item in agent 1's inventory to the user with context x^*, instead of recommending the two frequently co-purchased items together. Expected total reward of the "oracle" optimal policy, total rewards of CBMR-d and CBMR-ind, and the number of trainings of CBMR-d and CBMR-ind are shown in Table III for agent 1. We have $|\mathcal{L}_1| = 58$ and $|\mathcal{J}_1| = 14$.

We see that the total reward of CBMR-ind is higher than the total reward of CBMR-d for this case. This is due to the fact that CBMR-d spends more than double the time CBMR-ind spends in training and exploration phases since it trains and explores each action in $\mathcal{L}_1$ separately. The time averaged regrets of CBMR-d and CBMR-ind are given in Fig. 5. It is observed that CBMR-ind performs better than CBMR-d in all time slots,

TABLE III
TOTAL REWARDS OF THE "ORACLE" OPTIMAL POLICY, CBMR-D AND CBMR-IND, AND NUMBER OF TRAININGS OF CBMR-D AND CBMR-IND FOR GROUP-DEPENDENT PURCHASE PROBABILITIES

	Optimal	CBMR-d	CBMR-ind
Total Reward	11000	8724	9485
Trainings	-	12391	5342

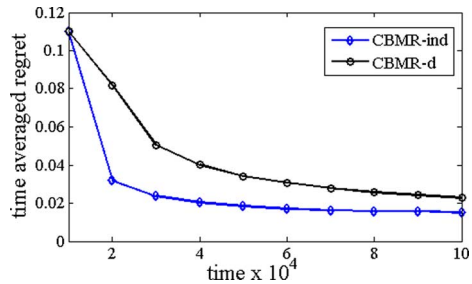


Fig. 5. Time averaged regrets of CBMR-d and CBMR-ind for group-dependent purchase probabilities.

TABLE IV
TOTAL REWARD OF AGENT 1 AS A FUNCTION OF THE COMMISSION IT CHARGES TO OTHER AGENTS

Commission c	0	0.1	0.2	0.3	0.4	0.5
Reward (CBMR-ind)	10471	10422	11476	12393	13340	14249

and the time averaged regret goes to 0. From these results it seems that CBMR-ind is a good alternative to CBMR-d even for group-dependent purchase probabilities.

C. Effect of Commission on the Performance

In this subsection we numerically simulate the effect of commissions $c_{1,j}$ that agent 1 charges to other agents on the total reward of agent 1. We consider CBMR-ind for independent purchase probabilities. We assume that agent 1 has one of the frequently co-purchased items for context x^*, while agent 3 has the other frequently co-purchased item. The total reward of agent 1 as a function of the commissions $c_{1,2} = c_{1,3} = c$ is given in Table IV. We note that there is no significant difference in the total reward when the commission is from 0 to 0.1. This is because 0.1 commission is not enough to incentivize agent 1 to recommend other agent's items to its users. However, for commissions greater than 0.1, the optimal policy recommends the two frequently co-purchased items together, hence agent 1 learns that it should get recommendations from agent 3. Therefore, after 0.1 the total reward of the agent is increasing in the commission. Another remark is that $c = 0$ corresponds to the case when agent 1 is not connected to the other agents. Therefore, this example also illustrates how the rewards change as network connectivity changes. Since prices of items are set to 1 in this section, the commission agent 1 charges can increase up to 1. But if the prices are different, then the commission cannot exceed the recommended item's price. In theory, in order to maximize its total reward from its own users, agent i can adaptively select its commissions $c_{i,j}$, $j \in \mathcal{M}_{-i}$ based on what it learned about the purchase probabilities. CBMR-d and CBMR-ind can be modified to adaptively select the commissions. Due to the limited space, we leave this as a future research topic.

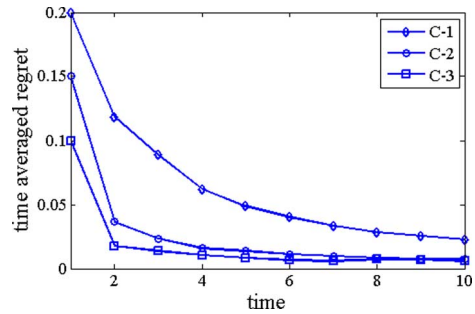


Fig. 6. Time averaged regret of CBMR-ind for independent purchase probabilities when agent 1 has both frequently co-purchased items (C-1), only one of the frequently co-purchased items (C-2) and none of the frequently co-purchased items (C-3).

D. Effect of the Set of Items of Each Agent on the Performance

In this subsection we compare three cases for independent purchase probabilities when agents use CBMR-ind. In C-1 agent 1 has both items that are frequently co-purchased in context x^*, in C-2 it has one of the items that is frequently co-purchased in context x^*, and in C-3 it has none of the items that are frequently co-purchased in context x^*. The total reward of agent 1 for these cases is 17744, 14249 and 9402 respectively, while the total expected reward of the optimal policy is 20000, 15000 and 10000 respectively. Note that the total reward for C-3 is almost half of the total reward for C-1 since the commission agent 1 gets for a frequently co-purchased item is 0.5. The time averaged regret of CBMR-ind for all these cases is given in Fig. 6. We see that the convergence rate for C-1 is slower than C-2 and C-3. This is due to the fact that in all of the training slots in C-1 a suboptimal set of items is recommended, while for C-2 and C-3 in some of the training slots the optimal set of items is recommended.

VII. CONCLUSION

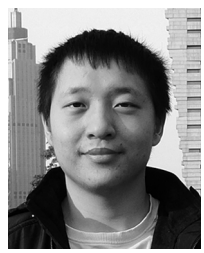
In this paper we have presented a novel set of algorithms for multi-agent learning within a decentralized social network, and characterized the effect of different network structures on performance. Our algorithms are able to achieve sublinear regret in all cases, with the regret being much smaller if the user's acceptance probabilities for different items are independent. By cooperating in a decentralized manner and using our algorithms, agents have the benefit of retaining their autonomy and privacy while still achieving near optimal performance.

REFERENCES

[1] K.-C. Chen, M. Chiang, and H. Poor, "From technological networks to social networks," *IEEE J. Sel. Areas Commun.*, vol. 31, no. 9, pp. 548–572, Sep. 2013.
[2] V. Krishnamurthy, "Quickest detection POMDPs with social learning: Interaction of local and global decision makers," *IEEE Trans. Inf. Theory,* , vol. 58, no. 8, pp. 5563–5587, Aug. 2012.
[3] V. Krishnamurthy and H. V. Poor, "Social learning and Bayesian games in multiagent signal processing: How do local and global decision makers interact?," *IEEE Signal Process. Mag.*, vol. 30, no. 3, pp. 43–57, May 2013.
[4] A. Jadbabaie, P. Molavi, A. Sandroni, and A. Tahbaz-Salehi, "Non-Bayesian social learning," *Games Econ. Behavior*, vol. 76, no. 1, pp. 210–225, 2012.
[5] A. Nedic and A. Ozdaglar, "Distributed subgradient methods for multi-agent optimization," *IEEE Trans. Autom. Control*, vol. 54, no. 1, pp. 48–61, Jan. 2009.

[6] A. Slivkins, "Contextual bandits with similarity information," in *Proc. 24th Annu. Conf. Learn. Theory (COLT)*, Jul. 2011.
[7] M. Dudik, D. Hsu, S. Kale, N. Karampatziakis, J. Langford, L. Reyzin, and T. Zhang, Efficient optimal learning for contextual bandits ArXiv, 2011 [Online]. Available: arXiv:1106.2369, to be published
[8] 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.
[9] W. Chu, L. Li, L. Reyzin, and R. E. Schapire, "Contextual bandits with linear payoff functions," in *Proc. Int. Conf. Artif. Intell. Statist. (AISTATS)*, 2011.
[10] C. Tekin and M. van der Schaar, "Distributed online Big Data classification using context information," in *Proc. 51st Allerton Conf. Commun., Control, Comput.*, Oct. 2013.
[11] 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, Oct. 2012.
[12] 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.
[13] P. Auer, N. Cesa-Bianchi, and P. Fischer, "Finite-time analysis of the multiarmed bandit problem," *Mach. Learn.*, vol. 47, pp. 235–256, 2002.
[14] K. Crammer and C. Gentile, "Multiclass classification with bandit feedback using adaptive regularization," *Mach. Learn.*, vol. 90, no. 3, pp. 347–383, 2013.
[15] A. Anandkumar, N. Michael, and A. Tang, "Opportunistic spectrum access with multiple users: Learning under competition," in *Proc. 29th IEEE Conf. Comput. Commun. (INFOCOM)*, Mar. 2010, pp. 1–9.
[16] K. Liu and Q. Zhao, "Distributed learning in multi-armed bandit with multiple players," *Signal Process., IEEE Trans.*, vol. 58, no. 11, pp. 5667–5681, Nov. 2010.
[17] C. Tekin and M. Liu, "Online learning of rested and restless bandits," *IEEE Trans. Inf. Theory*, vol. 58, no. 8, pp. 5588–5611, Aug. 2012.
[18] 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, Mar. 2013.
[19] C. Tekin and M. Liu, "Online learning in decentralized multi-user spectrum access with synchronized explorations," in *Proc. IEEE Military Commun. Conf. (MILCOM)*, 2012, pp. 1–6.
[20] Y. Deshpande and A. Montanari, "Linear bandits in high dimension and recommendation systems," in *Proc. 50th Annu. Allerton Conf. Commun., Control, Comput.*, 2012, pp. 1750–1754.
[21] P. Kohli, M. Salek, and G. Stoddard, "A fast bandit algorithm for recommendations to users with heterogeneous tastes," in *Proc. 27th AIII Conf. Artif. Intell.*, Jul. 2013.
[22] T. Hofmann, "Latent semantic models for collaborative filtering," *ACM Trans. Inf. Syst. (TOIS)*, vol. 22, no. 1, pp. 89–115, 2004.
[23] N. Sahoo, P. V. Singh, and T. Mukhopadhyay, "A hidden Markov model for collaborative filtering," *MIS Quarterly*, vol. 36, no. 4, pp. 1329–1356, 2012.
[24] G. Linden, B. Smith, and J. York, "Amazon.com recommendations: Item-to-item collaborative filtering," *IEEE Internet Comput.*, vol. 7, no. 1, pp. 76–80, Jan./Feb. 2003.
[25] U. Panniello, A. Tuzhilin, and M. Gorgoglione, "Comparing context-aware recommender systems in terms of accuracy and diversity," *User Model. User-Adapt. Interact.*, pp. 1–31, 2012.
[26] K. Miyahara and M. J. Pazzani, "Collaborative filtering with the simple Bayesian classifier," in *Proc. PRICAI Topics Artif. Intell.*, 2000, pp. 679–689.
[27] M. O'Connor and J. Herlocker, "Clustering items for collaborative filtering," in *Proc. ACM SIGIR Workshop Recomm. Syst.*, 1999, vol. 128.
[28] G. Shani, R. I. Brafman, and D. Heckerman, "An MDP-based recommender system," in *Proc. 18th Conf. Uncert. Artif. Intellig.*, 2002, pp. 453–460.
[29] C.-N. Ziegler, "Towards decentralized recommender systems," Ph.D. dissertation, Albert-Ludwigs-Universitat Freiburg, Freiburg im Breisgau, Germany, 2005.
[30] M. Balabanović and Y. Shoham, "Fab: Content-based, collaborative recommendation," *Commun. ACM*, vol. 40, no. 3, pp. 66–72, 1997.
[31] C. Tekin, S. Zhang, and M. van der Schaar, "Distributed online learning in social recommender systems – Online appendix," ArXiv, 2013 [Online]. Available: arXiv:1309.6707, to be published
[32] C. Tekin and M. van der Schaar, "Distributed online learning via cooperative contextual bandits," ArXiv, 2013 [Online]. Available: arXiv:1308.4568, to be published
[33] J. Leskovec, L. A. Adamic, and B. A. Huberman, "The dynamics of viral marketing," *ACM Trans. Web (TWEB)*, vol. 1, no. 1, p. 5, 2007.

Cem Tekin is a Postdoctoral Scholar at University of California, Los Angeles. He 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. His research interests include machine learning, multi-armed bandit problems, data mining, multi-agent systems and game theory.



Simpson Zhang is a Ph.D. candidate in the UCLA Department of Economics. He received his Bachelors degree from Duke University with a double major in math and economics. His current research focuses on reputational mechanisms, multi-armed bandit problems, and network formation and design.

Mihaela van der Schaar 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.