Contextual Combinatorial Multi-output GP Bandits with Group Constraints

Anonymous authors
Paper under double-blind review

Abstract

In federated multi-armed bandit problems, maximizing global reward while satisfying minimum privacy requirements to protect clients is the main goal. To formulate such problems, we consider a combinatorial contextual bandit setting with groups and changing action sets, where similar base arms arrive in groups and a set of base arms, called a super arm, must be chosen in each round to maximize super arm reward while satisfying the constraints of the rewards of groups from which base arms were chosen. To allow for greater flexibility, we let each base arm have two outcomes, modeled as the output of a two-output Gaussian process (GP), where one outcome is used to compute super arm reward and the other for group reward. We then propose a novel double-UCB GP-bandit algorithm, called Thresholded Combinatorial Gaussian Process Upper Confidence Bounds (TCGP-UCB), which balances between maximizing cumulative super arm reward and satisfying group reward constraints and can be tuned to prefer one over the other. We also define a new notion of regret that combines super arm regret with group reward constraint satisfaction and prove that TCGP-UCB incurs $\tilde{O}(\sqrt{KT\overline{\gamma}_T})$ regret with high probability, where $\overline{\gamma}_T$ is the maximum information gain associated with the set of base arm contexts that appeared in the first T rounds and K is the maximum super arm cardinality over all rounds. We lastly show in experiments based on federated learning setups that our algorithm accumulates a reward comparable with that of the state-of-the-art combinatorial bandit algorithm while satisfying group constraints hence privacy requirements.

1 Introduction

The multi-armed bandit problem is a prominent example of reinforcement learning that studies the sequential interaction of a learner with its environment under partial feedback (Robbins, 1952). Out of the many variants of the multi-armed bandit problem, the combinatorial and contextual bandits have been thoroughly investigated due to their rich set of real-life applications. In combinatorial bandits, the learner selects a super arm, which is a subset of the available base arms in each round (Chen et al., 2013; Cesa-Bianchi & Lugosi, 2012). In the semi-bandit feedback model, at the end of the round, the learner observes the outcomes of the base arms in the selected super arm together with the super arm reward. In contextual bandits, at the beginning of every round, the learner observes side-information about the outcomes of the base arms available in that round before deciding which arm to select (Slivkins, 2011; Lu et al., 2010). In the changing action set variant, there is varying base arm availability in each round. Combinatorial bandits and contextual bandits are deployed in applications ranging from influence maximization in a social network to news article recommendation (Li et al., 2010; Chen et al., 2016b).

In this paper, we consider contextual combinatorial multi-armed bandits with changing action sets (C3-MAB), which models the repeated interaction between an agent and its dynamically changing environment. In each round t, the agent observes the available base arms and their contexts, selects a subset of the available base arms, which is called a super arm, collects a reward, and observes noisy outcomes of the selected base arms. We consider the scenario when the base arm availability in each round changes in an arbitrary fashion, known as the changing action sets setting. Therefore, we analyze the regret under any given sequence of base arm availabilities. On the other hand, given a particular base arm and its context, the outcome of the base arm is assumed to come from a fixed distribution parameterized by the arm's context. The goal is to maximize the cumulative reward in a given number of rounds without knowing future context arrivals and the function that maps the contexts to base arm outcomes. Achieving this goal requires careful tuning of exploration and exploitation by adapting decisions in real time based on the problem structure and past history. with group constraints. What is new in our setup is the concept of groups, which are unions of base arms that share a common feature. In every round t, the learner is able to observe the available base arms along with their contexts and selects a super arm, from which the noisy base arm outcomes are observed and the reward is collected. Due to the varying availability of base arms, available super arms and groups change from round to round. We also consider the notion of satisfaction (Reverdy et al., 2017) in relation to set of base arms, called groups. The goal is then to maximize the cumulative super arm reward while ensuring that the groups comprising the super arm satisfy their constraints.

It is essential to place assumptions on the function mapping contexts to outcomes (i.e., the function that needs to be learned, say f). While other C3-MAB works such as (Nika et al., 2020; Chen et al., 2016a; 2018) place explicit smoothness restrictions on f, we make use of Gaussian processes (GP), which induce a smoothness assumption. Moreover, the posterior mean and variance resulting from GP gives us high probability confidence bounds on the expected outcome function which leads to a desirable balance between exploration and exploitation (Srinivas et al., 2012).

Our work, and specifically introduction of groups, is motivated by the need to balance between privacy and reward in a federated learning setup. Federated learning (FL) is a distributed machine learning method that allows for training a model across a decentralized ensemble of clients. In FL, clients train a model on their local data and then send their local model to a centralized server. This helps protect the privacy of clients as they do not share their data directly with the server (McMahan et al., 2017). Our work introduces a setup that can handle a privacy aware FL setup, where clients have privacy requirements on the maximum amount of data leakage through local model or weight transfer to the server. This is important, as some portions of a client's training data can be retrieved just from their shared gradients with the server (Melis et al., 2019; Chen & Campbell, 2021).

In our setup, base arms represent client-request pairs, groups are clients, and super arms are subsets of client-request pairs. For each such pair, the context is related to the amount of information that the client is to share with the server, specifically, it is the used data set percentage which determines the privacy level of the client. Our aim is to satisfy clients' privacy requirements while maximizing the information obtained from the client's trained model. There is a trade-off between these two goals as using a higher percentage of data in training leads to less privacy for clients due to information leakage but a greater amount of information is retrieved from the data by the server. This is also mentioned in (Truong et al., 2021) as a trade-off between efficiency and privacy guarantee and in (Yang et al., 2019) as a trade-off between learning performance and privacy of clients.

This trade-off has motivated us to model the expected outcome function as a sample from a two-output GP, in order to make use of the inherent correlation between privacy leakage and the information retrieved. The details of how a two-output GP helps us as well as a detailed explanation of the application of our problem formulation to a privacy aware federated learning is given in Section 2.8.

1.1 Our contributions

Our contributions are: **(i)** We propose a new C3-MAB problem with groups and group constraints that is applicable to real-life problems, including a privacy-aware federated learning setup where the goal is to maximize overall reward while respecting users' varying privacy requirements. **(ii)** We propose a new notion of regret called group regret, and define total regret as a weighted combination of group and super arm regret in terms of $\zeta \in [0, 1]$. **(iii)** We use a two-output GP which allows for modeling setups where overall reward and group reward are functions of two different but correlated base arm outcomes. **(iv)** We use a double UCB approach in our algorithm by having different exploration bonuses for groups and super arms which is dependent on a tunable parameter. **(v)** We derive an information theoretic regret bound for our algorithm given by $\tilde{O}(\sqrt{KT\overline{\gamma}_T})$. By assuming a fixed cardinality for every super arm, we express our regret bound in terms of the classical maximum information gain γ_T which is given as $\tilde{O}(\sqrt{KT\gamma_T})$.

Table 1: Comparison of our problem setting with previous works (CC stands for contextual-changing action sets).

Work	**Context space**	**CC**	**Smoothness**	**Groups**	**Group thresholds**
(Chen et al., 2013)	Finite	No	Explicit	No	No
(Chen et al., 2018)	Infinite	Yes	Explicit	No	No
(Krause & Ong, 2011)	Compact	No	GP-induced	No	No
(Nika et al., 2020)	Compact	Yes	Explicit	No	No
This work	Compact	Yes	GP-induced	Yes	Yes

1.2 Related Work

The contextual combinatorial multi-armed bandit problem (CC-MAB) has been investigated thoroughly in recent years (Li et al., 2016; Qin et al., 2014). In the work of (Li et al., 2016), the learner can observe the reward of the super arm and the rewards of a subset of the base arms in the super arm selected due to cascading feedback according to some stopping criteria. In the work of (Qin et al., 2014), the contextual combinatorial MAB problem is applied to online recommendation.

The contextual combinatorial multi-armed bandits with changing action sets problem has been explored in (Chen et al., 2018; Nika et al., 2020). In the work of (Chen et al., 2018), the super arm reward is submodular and since the context space is infinite, they form a partition of the context space with hypercubes depending on context information thus addressing the varying availability of arms by exploiting the similarities between contexts in the same hypercube. The work done by (Nika et al., 2020) makes use of adaptive discretization instead of fixed discretization which addresses the limited similarity information of the arms that can be gathered by fixed discretization.

GP bandits have been utilized in the contextual MAB setup as well as for adaptive discretization (Krause & Ong, 2011; Shekhar et al., 2018). While adaptive discretization has been shown to be working well in large context spaces and setups with changing action sets, in the case where the number of base arms is finite, assuming the expected base arm outcomes are a sample from a GP could be a better approach as it eliminates the need for the explicit assumption that the expected base arm outcomes are Lipschitz continuous. In this paper we also use the smoothness induced by a GP instead of performing explicit discretization. A comparison of our problem setup with other similar works can be found in Table 1.

The thresholding multi-armed bandit problem has been studied in (Locatelli et al., 2016; Mukherjee et al., 2017). These works consider thresholding as having the mean of a base arm be above a certain value. In (Reverdy et al., 2017), there is the notion of satisficing instead of thresholding, which is a combination of satisfaction, which is the learner's desire to have a reward above a threshold and sufficiency, meaning to have satisfaction for base arms at a certain level of confidence. To our knowledge, there is no previous research which deals with group thresholding or a thresholding setup within the C3-MAB problem.

Federated learning was proposed by (McMahan et al., 2017) as a technique to address privacy and security concerns regarding centralized training performed in a main server by collecting private data from the clients. To solve this issue, they propose the FedAvg algorithm that performs local stochastic gradient descent for each client and averages the models from the clients on the main server to update the global model. The work in (Wei et al., 2020) discusses information leakage in federated learning and presents the algorithm NbAFL that works based on the concept of differential privacy by adding artificial noises to the parameters of the clients before aggregating the model. Note that although our setup is motivated by privacy aware FL, our work is not positioned in the FL space and thus our algorithm competes with C3-MAB algorithms and not the likes of NbAFL.

2 Problem Formulation

2.1 Base Arms and Base Arm Outcomes

The sequential decision-making problem proceeds over T rounds indexed by $t \in [T]$. In each round t, M_t base arms indexed by the set $\mathcal{M}_t = [M_t]$ arrive. Cardinality of the set of available base arms in any round is bounded above by $M < \infty$. Each base arm $m \in \mathcal{M}_t$ comes with a context $x_{t,m}$ that resides in the context set $\mathcal{X}$. The set of available contexts in round t is denoted by $\mathcal{X}_t = \{x_{t,m}\}_{m \in \mathcal{M}_t}$.

When selected, a base arm with context x yields a two-dimensional random outcome $\boldsymbol{r}(x) \in \mathbb{R}^2$. The two-dimentionality of the outcome will be motivated by groups in Section 2.2. Then, the expected reward function that is unknown to the learner is represented by $\boldsymbol{f} : \mathcal{X} \to \mathbb{R}^2$, whose form will be described in detail in Section 2.5. We then define the random outcome as $\boldsymbol{r}(x) = \boldsymbol{f}(x) + \boldsymbol{\eta}$ where $\boldsymbol{\eta} \sim \mathcal{N}(\boldsymbol{0}, \sigma^2 \boldsymbol{I})$ represents the two-dimensional observation noise that is independent across base arms and rounds. In our motivating example of federated learning, base arms represent client-request pairs while the first base arm outcome is the amount of information that the server is able to retrieve from the client while the second one is the information leakage probability associated with each client-request pair. The context associated with each base arm corresponds to its data set usage percentage.

2.2 Super Arm and Group Rewards

For any function h, which takes a single context as an argument, given a k-tuple of contexts $\boldsymbol{x} = [x_1, \ldots, x_k]$, we let $\boldsymbol{h}(\boldsymbol{x}) = [\boldsymbol{h}(x_1), \ldots, \boldsymbol{h}(x_k)]$, where $\boldsymbol{h} \colon \mathcal{X} \to \mathbb{R}^d$ is a function for some $d \geq 2$. Moreover, we let $\boldsymbol{h}_i(\boldsymbol{x}) = [h_i(x_1), \ldots, h_i(x_k)]$, where the subscript $1 \leq i \leq d$ indicates the i^{th} element. Equipped with this notation, at each round t, super arms are subsets of available base arms and the chosen super arm is S_t. We denote by $\mathcal{S}_t$ the set of feasible super arms in round t and by $\mathcal{S} = \cup_{t \geq 1} \mathcal{S}_t$ the overall feasible set of super arms. We assume that the maximum number of base arms in a super arm does not exceed a fixed $K \in \mathbb{N}$ that is known to the learner. That is, for any $S \in \mathcal{S}$, we have, $|S| \leq K$.

The reward of a super arm $S \in \mathcal{S}$ depends on the outcomes of base arms in S. We represent the reward of a super arm S with context vector $\boldsymbol{x}_{t,S}$ by random variable $U(S, r_1(\boldsymbol{x}_{t,S}))$, where U is a deterministic function. Moreover, we have that the expected super arm reward is a function of only the set of base arms in S and their expected outcome vector, given by $u(S, f_1(\boldsymbol{x}_{t,S})) = \mathbb{E}[U(S, r_1(\boldsymbol{x}_{t,S}))|\boldsymbol{f}]$.[1] We assume that for all $S \in \mathcal{S}$, the expected super arm reward function u is monotonically non-decreasing with respect to the expected outcome vector. We also assume that u varies smoothly as a function of expected base arm outcomes. Both assumptions are standard in the C3-MAB literature (Nika et al., 2020) and are stated formally below.

Assumption 1 (Monotonicity) *For all $S \in \mathcal{S}$ and for any $\boldsymbol{f} = [f_1, \ldots, f_{|S|}]^T \in \mathbb{R}^{|S|}$ and $\boldsymbol{g} = [g_1, \ldots, g_{|S|}]^T \in \mathbb{R}^{|S|}$, if $f_m \leq g_m$, $\forall m \leq |S|$, then $u(S, \boldsymbol{f}) \leq u(S, \boldsymbol{g})$.*

Assumption 2 (Lipschitz continuity) *For all $S \in \mathcal{S}$, there exists $B' > 0$ such that for any $\boldsymbol{f} = [f_1, \ldots, f_{|S|}]^T \in \mathbb{R}^{|S|}$ and $\boldsymbol{g} = [g_1, \ldots, g_{|S|}]^T \in \mathbb{R}^{|S|}$, we have $|u(S, \boldsymbol{f}) - u(S, \boldsymbol{g})| \leq B' \sum_{i=1}^{|S|} |f_i - g_i|$.*

Groups are defined as the sets of base arms (i.e., client-request pairs) from the same client and they have round-varying cardinalities due to changing action sets. We denote by $\mathcal{G}_t$ the set of feasible groups in round t and the reward of a group $G \in \mathcal{G}_t$ depends on the second outcome of the base arms in $G \cap S_t$. Let $\boldsymbol{x}_{t,G}$ represent the vector of contexts of base arms in G.[2] We represent the reward of a group G by the random variable $V_G(G \cap S_t, r_2(\boldsymbol{x}_{t,G \cap S_t}))$. Notice that the reward of a group depends on the observed base arms since the observed base arm outcomes in a group are ones that belong to the chosen super arm. We stress that V_G here is a deterministic function of its argument and the randomness comes from r_2. As typical in the CMAB literature, given the base arm outcome function $\boldsymbol{f}$, we assume that expected group reward is a function of only the set of base arms in $G \cap S_t$ and their expected outcome vector. Therefore, we define $v_G(G \cap S_t, f_2(\boldsymbol{x}_{t,G \cap S_t})) = \mathbb{E}[V_G(G \cap S_t, r_2(\boldsymbol{x}_{t,G \cap S_t}))|\boldsymbol{f}]$. Note that V_G and v_G are vector functions and

[1] We will suppress S from $U(S, \ldots)$ and $u(S, \ldots)$ when S is clear from context.
[2] We will drop t from $\boldsymbol{x}_{t,G}$ when clear from context.

take as input a $|G \cap S_t|$ dimensional vector.[3] Lastly, similar to the super arm reward function, we impose monotonicity and Lipschitz continuity assumptions on the group reward function.

Assumption 3 (Monotonicity) *For all $A \subseteq G$, for all $G \in \mathcal{G}_t$, and for all $t \in [T]$ and for any $\boldsymbol{f} = [f_1, \dots, f_{|A|}]^T \in \mathbb{R}^{|A|}$ and $\boldsymbol{g} = [g_1, \dots, g_{|A|}]^T \in \mathbb{R}^{|A|}$, if $f_m \leq g_m$, $\forall m \leq |A|$, then $v_G(\boldsymbol{f}) \leq v_G(\boldsymbol{g})$.*

Assumption 4 (Lipschitz continuity) *For all $A \subseteq G$, for all $G \in \mathcal{G}_t$ and for all $t \in [T]$, there exists $B_A > 0$ such that for any $\boldsymbol{f} = [f_1, \dots, f_{|A|}]^T \in \mathbb{R}^{|A|}$ and $\boldsymbol{g} = [g_1, \dots, g_{|A|}]^T \in \mathbb{R}^{|A|}$, we have $|v_G(\boldsymbol{f}) - v_G(\boldsymbol{g})| \leq B_A \sum_{i=1}^{|A|} |f_i - g_i|$.*

We define $B := \max_{\forall A \subseteq G, \forall G \in \mathcal{G}_t \ \forall t \in [T]} B_A$, which will be later used in proofs. In federated learning, group reward is related to the information leakage probability of each base arm in a group. Therefore, it represents the privacy level of a client.

In line with prior work (Nika et al., 2021), we assume that the learner knows u and v_G for all G perfectly, but does not know $\boldsymbol{f}$ beforehand. We propose an extended semi-bandit feedback model, where when super arm S_t is selected in round t, the learner observes $U(r_1(\boldsymbol{x}_{S_t}))$, $V_G(r_2(\boldsymbol{x}_{G \cap S_t}))$ for $G \in \mathcal{G}_t$ and $\boldsymbol{r}(x_{t,m})$ for $m \in S_t$ at the end of the round.

2.3 Regret

Our learning objective is to choose super arms that yield maximum rewards while ensuring that the groups containing the base arms of the chosen super arms satisfy a certain level of quality, which is characterized by the group threshold. In accord with our goal, we define regret notions for both super arms and groups. In our regret analysis, as typical in contextual bandits (Nika et al., 2020), we assume that the sequence of available base arms $\{\mathcal{X}_t\}_{t=1}^T$ is fixed, hence context arrivals are not affected by past actions. We assume that each group $G \in \mathcal{G}_t$ has a threshold $\gamma_{t,G}$, which represents the minimum reward that the group requires when one or more of its forming base arms are in the chosen super arm. For instance, in federated learning, $\gamma_{t,G}$ represents the minimum desired privacy level of each client, specifically the probability that the client's data does not leak. This is important as we want to maintain privacy for clients during training. We say that group $G \in \mathcal{G}_t$ satisfies its threshold when $v_G(f_2(\boldsymbol{x}_{G \cap S_t})) \geq \gamma_{t,G}$ and we denote the set of feasible groups who do not satisfy their thresholds by $\tilde{\mathcal{G}}_t$. We define the group regret as

$$R_g(T) = \sum_{t=1}^{T} \sum_{G \in \tilde{\mathcal{G}}_t} [\gamma_{t,G} - v_G(f_2(\boldsymbol{x}_{G \cap S_t}))]_+,$$

where $[\cdot]_+ = \max\{\cdot, 0\}$ and for any $G \in \tilde{\mathcal{G}}_t$, it holds that $[\gamma_{t,G} - v_G(f_2(\boldsymbol{x}_{G \cap S_t}))]_+ = \gamma_{t,G} - v_G(f_2(\boldsymbol{x}_{G \cap S_t}))$.

Let $\mathcal{S}'_t \subseteq \mathcal{S}_t$ represent the set of super arms whose forming base arms' membering groups satisfy their thresholds, i.e., $\{S \in \mathcal{S}_t : \forall (G \in \mathcal{G}_t), v_G(f_2(\boldsymbol{x}_{t,G \cap S})) \geq \gamma_{t,G}\}$. The super arm regret is defined as the standard α-approximation regret in CMAB:

$$R_s(T) = \alpha \sum_{t=1}^{T} \mathrm{opt}(f_t) - \sum_{t=1}^{T} u(f_1(\boldsymbol{x}_{t,S_t})),$$

where $\mathrm{opt}(f_t) = \max_{S \in \mathcal{S}'_t} u(f_1(\boldsymbol{x}_{t,S}))$. Note that any optimal super arm $S_t^* \in \arg\max_{S \in \mathcal{S}'_t} u(f_1(\boldsymbol{x}_{t,S}))$ is restricted to satisfy all its corresponding groups and this will be further explained in Section 2.4.2. Otherwise, the policy which always selects the optimal super arms will incur linear group regret.

The total regret is defined as a weighted combination of group and super arm regrets. Given the trade-off parameter $\zeta \in [0, 1]$, we define it as

$$R(T) = \zeta R_g(T) + (1 - \zeta) R_s(T) \ . \tag{1}$$

Setting $\zeta = 1$ reduces the problem to a combinatorial version of satisfying, while setting $\zeta = 0$ reduces the problem to regret minimization in standard CMAB (Chen et al., 2013).

[3] We will suppress $G \cap S_t$ from $V_G(G \cap S_t, \dots)$ and $v_G(G \cap S_t, \dots)$ when clear from context.

2.4 Computation Oracles

In the traditional C3-MAB setting where group constraints are not considered, there exists one optimization problem, namely identifying a super arm to play in each round (Chen et al., 2016a; Nika et al., 2020). However, in our new setup where group constraints are considered, an additional optimization problem needs to be solved to identify super arms whose comprising groups satisfy their constraints, from here on called *good* groups.

2.4.1 Identifying good groups

Note that in our setup, a feasible super arm can be any subset of available base arms in its respective round, thus it need not contain all members of any group. For instance, given base arms $\mathcal{M} = [5]$ and groups $\mathcal{G} = \{G_1, G_2, G_3\} = \{\{1,4\},\{2,4\},\{1,3,5\}\}$, then $S = \{2,3,5\}$ can be a feasible super arm. In this example, only the second and third groups will need to satisfy their constraints. More specifically, only their intersections with the super arm S need to be checked. Thus, we need to check $v_{G_2}(f_2(\boldsymbol{x}_{G_2 \cap S})) = v_{G_2}(f_2(\boldsymbol{x}_{\{2\}})) \geq \gamma_{G_2}$ and $v_{G_3}(f_2(\boldsymbol{x}_{G_3 \cap S})) = v_{G_3}(f_2(\boldsymbol{x}_{\{3,5\}})) \geq \gamma_{G_3}$.

Then, considering our assumption of group rewards being monotone increasing, given in Assumption 3, determining super arms whose comprising groups satisfy their constraints is equivalent to determining subsets of groups who satisfy the constraint. Given that $\hat{f}_2$ is an approximation of the expected group outcome function, f_2, then the good group identification problem in round t can be expressed formally as identifying the set $\mathcal{G}_{t,\text{good}}$ defined as $\mathcal{G}_{t,\text{good}} := \{G' \subseteq G \mid G \in \mathcal{G}_t \text{ and } v_G(\hat{f}_2(\boldsymbol{x}_{G'})) \geq \gamma_{t,G}\}$. Notice that $\mathcal{G}_{t,\text{good}}$ contains not just good groups, but also good subsets of groups (i.e., subsets of groups whose constraint is satisfied). We make use of an exact oracle that returns the set $\mathcal{G}_{t,\text{good}}$ given estimator $\hat{f}_2$, called $\text{Oracle}_{\text{grp}}$. Thus, in round t, we have $\mathcal{G}_{t,\text{good}} = \text{Oracle}_{\text{grp}}(\hat{f}_{t,2})$.

2.4.2 Identifying optimal super arm

Once $\mathcal{G}_{t,\text{good}}$ is identified, the next task is to identify the super arm that yields the highest reward. Using the set of good groups returned by $\text{Oracle}_{\text{grp}}$, the optimization problem of finding the optimal super arm in round t, S_t, given estimates of the expected super arm outcome function, $\hat{f}_1$, can be written as $S_t = \arg\max_S \{u(\hat{f}_1(\boldsymbol{x}_{t,S})) \mid S \in \mathcal{S}_t \text{ and } \forall G \in \mathcal{G}_t,\ S \cap G \in \mathcal{G}_{t,\text{good}}\}$. In other words, S_t is the super arm that yields the highest reward calculated using $\hat{f}_1$ and whose intersections with any group satisfies that group's constraints. Then, we make use of an α-approximate oracle, called $\text{Oracle}_{\text{spr}}$, where $u(f(\boldsymbol{x}_{t,\text{Oracle}_{\text{spr}}(\hat{f}_{t,1})})) \geq \alpha \times \text{opt}(\hat{f}_{t,1})$. Our algorithm, which will be described in detail in Section 3, will make use of both oracles. Lastly, it should be noted that both oracles are deterministic given their inputs.

2.5 Structure of Base Arm Outcomes

To ensure that the learner performs well, some regularity conditions are necessary on $\boldsymbol{f}$. For our paper, we model $\boldsymbol{f}$ as a sample from a two-output GP, defined below.

Definition 1. *A two-output Gaussian Process with index set $\mathcal{X}$ is a collection of 2-dimensional random variables $(\boldsymbol{f}(x))_{x \in \mathcal{X}}$ which satisfy the condition that $(\boldsymbol{f}(x_1), \ldots, \boldsymbol{f}(x_n))$ has a multivariate normal distribution for all $(x_1, \ldots, x_n)$ and $n \in \mathbb{N}$. The probability law of the GP is governed by its vector-valued mean function given by $x \mapsto \boldsymbol{\mu}(x) = \mathbb{E}[\boldsymbol{f}(x)] \in \mathbb{R}^2$ and its matrix-valued covariance function given by $(x_1, x_2) \mapsto \boldsymbol{k}(x_1, x_2) = \mathbb{E}[(\boldsymbol{f}(x_1) - \boldsymbol{\mu}(x_1))(\boldsymbol{f}(x_2) - \boldsymbol{\mu}(x_2))^T] \in \mathbb{R}^{2\times 2}$.*

We assume that we have bounded variance, that is, $k_{jj}(x,x) \leq 1$ for every $x \in \mathcal{X}$ and $j \in \{1,2\}$. This is a standard assumption generally used in GP bandits (Srinivas et al., 2012).

2.6 Posterior Distribution of Base Arm Outcomes

Our learning algorithm will make use of the posterior distribution of the two-output GP-sampled function $\boldsymbol{f}$. Given a fixed $N \in \mathbb{N}$ we consider a finite sequence $\tilde{\boldsymbol{x}}_{[N]} = [\tilde{x}_1, \ldots, \tilde{x}_N]^T$ of contexts with corresponding outcome vector ($2N$-dimensional) $\boldsymbol{r}_{[N]} := [\boldsymbol{r}(\tilde{x}_1)^T, \ldots, \boldsymbol{r}(\tilde{x}_N)^T]^T$ and the corresponding expected outcome vector ($2N$-dimensional) $\boldsymbol{f}_{[N]} = [\boldsymbol{f}(\tilde{x}_1)^T, \dots, \boldsymbol{f}(\tilde{x}_N)^T]^T$. For every $n \leq N$, we have $\boldsymbol{r}(\tilde{x}_n) = \boldsymbol{f}(\tilde{x}_n) + \boldsymbol{\eta}_n$ where $\boldsymbol{\eta}_n$ is the noise corresponding to that outcome. The posterior distribution of $\boldsymbol{f}$ given $\boldsymbol{r}_{[N]}$ is that of a two-output GP characterized by its mean $\boldsymbol{\mu}_N$ and its covariance $\boldsymbol{k}_N$ which are given as follows:

$$\boldsymbol{\mu}_N(\tilde{x}) = (\boldsymbol{k}_{[N]}(\tilde{x}))(\boldsymbol{K}_{[N]} + \sigma^2 \boldsymbol{I}_{2N})^{-1} \boldsymbol{r}_{[N]}^T,$$
$$\boldsymbol{k}_N(\tilde{x}, \tilde{x}') = \boldsymbol{k}(\tilde{x}, \tilde{x}') - (\boldsymbol{k}_{[N]}(\tilde{x}))(\boldsymbol{K}_{[N]} + \sigma^2 \boldsymbol{I}_{2N})^{-1} \boldsymbol{k}_{[N]}(\tilde{x}')^T.$$

Here $\boldsymbol{k}_{[N]}(\tilde{x}) = [\boldsymbol{k}(\tilde{x}, \tilde{x}_1), \dots, \boldsymbol{k}(\tilde{x}, \tilde{x}_N)] \in \mathbb{R}^{2 \times 2N}$ and

$$\boldsymbol{K}_{[N]} = \begin{bmatrix} \boldsymbol{k}(\tilde{x}_1, \tilde{x}_1), & \cdots, & \boldsymbol{k}(\tilde{x}_1, \tilde{x}_N) \\ \vdots & \ddots & \vdots \\ \boldsymbol{k}(\tilde{x}_N, \tilde{x}_1) & \cdots, & \boldsymbol{k}(\tilde{x}_N, \tilde{x}_N) \end{bmatrix}$$

Overall, the posterior of $\boldsymbol{f}(x)$ is given by $\mathcal{N}(\boldsymbol{\mu}_N(x), \boldsymbol{k}_N(x, x))$ and for each $j \in \{1, 2\}$, the posterior distribution of $f_j(x)$ is $\mathcal{N}(\mu_{j\,N}(x), (\sigma_{j\,N}(x))^2)$, where $(\sigma_{j\,N}(x))^2 = k_{jj\,N}(x, x)$. Moreover, the posterior distribution of the outcome $\boldsymbol{r}(x)$ is given by $\mathcal{N}(\boldsymbol{\mu}_N(x), \boldsymbol{k}_N(x, x) + \sigma^2 \boldsymbol{I}_2)$.

2.7 The Information Gain

Regret bounds for GP bandits depend on how well $\boldsymbol{f}$ can be learned from sequential interaction. The learning difficulty is quantized by the information gain, which accounts for the reduction in entropy of a random vector after a sequence of (correlated) observations. For a length N sequence of contexts $\tilde{\boldsymbol{x}}_{[N]}$, evaluations of $\boldsymbol{f}$ at contexts in $\tilde{\boldsymbol{x}}_{[N]}$, given by $\boldsymbol{f}_{[N]}$, and the corresponding sequence of outcomes $\boldsymbol{r}_{[N]}$, the information gain is defined as $I(\boldsymbol{r}_{[N]}; \boldsymbol{f}_{[N]}) := H(\boldsymbol{r}_{[N]}) - H(\boldsymbol{r}_{[N]} | \boldsymbol{f}_{[N]})$ where $H(\cdot)$ and $H(\cdot|\cdot)$ represent the entropy and the conditional entropy operators, respectively. Essentially, the information gain quantifies the reduction in the entropy of $\boldsymbol{f}_{[N]}$ given $\boldsymbol{r}_{[N]}$. The maximum information gain over the context set $\mathcal{X}$ is denoted by $\gamma_{j\,N} := \sup_{\tilde{\boldsymbol{x}}_{[N]}: \tilde{x}_i \in \mathcal{X}, i \in [N]} I_j(\boldsymbol{r}_{[N]}; \boldsymbol{f}_{[N]})$.

It is important to understand that $\gamma_{j\,N}$ is the maximum information gain for an N-tuple of contexts from $\mathcal{X}$. For large context spaces, $\gamma_{j\,N}$ can be very large. As we have varying base arm availability from round to round due to changing action sets, we only need to know the maximum information gain that is associated with a fixed sequence of base arm contexts arrivals. As we have a finite maximum information gain, this motivates us to relate the regret bounds to the informativeness of the available base arms. To adapt the definition of the maximum information gain to our setup with changing action sets and to ensure tight bounds, we define a new term $\overline{\gamma_{j\,T}}$. We let $\mathcal{Z}_t \subset 2^{\mathcal{X}_t}$ for $t \geq 1$ be the set of sets of context vectors which correspond to the base arms in the feasible set of super arms $\mathcal{S}_t$. We let $\tilde{\boldsymbol{z}}_t := \boldsymbol{x}_{t,S}$ be an element of $\mathcal{Z}_t$ for a super arm S and we let $\tilde{\boldsymbol{z}}_{[T]} := [\tilde{\boldsymbol{z}}_1^T, \dots, \tilde{\boldsymbol{z}}_T^T]$. Then the maximum information gain which is associated with the context arrivals $\mathcal{X}_1, \dots, \mathcal{X}_T$ is

$$\overline{\gamma_{j\,T}} = \max_{\tilde{\boldsymbol{z}}_{[T]}: \tilde{\boldsymbol{z}}_t \in \mathcal{Z}_t, t \leq T} I(\boldsymbol{r}_j(\tilde{\boldsymbol{z}}_{[T]}); \boldsymbol{f}_j(\tilde{\boldsymbol{z}}_{[T]})).$$

The information gain for any T element sequence of feasible super arms has a maximum given by $\overline{\gamma_{j\,T}}$. As it will be shown in Section 4, our regret bounds depend on $\overline{\gamma_{j\,T}}$.

2.8 Motivation

Our setup is as depicted in Figure 1 and it is a crowdsourcing application. At each round t, $|\mathcal{G}_t|$ groups, hence clients, are available. Each client has one or more requests and these client-request pairs correspond to base arms, $m \in \mathcal{M}_t$. Super arms are comprised of combinations of these pairs. The chosen super arm S_t includes the client-request pairs having bold arrows. These pairs also have dashed arrows as they train their data locally before sending their model back to the server. Base arm outcomes, namely $\boldsymbol{f}$, are the amount of information that the server retrieves and also the information leakage probability associated with each pair. The amount of retrieved information which is related to super arm reward (u) is aimed to be maximized, and the information leakage probability which is inversely related to the group reward (v_G) is aimed to satisfy a

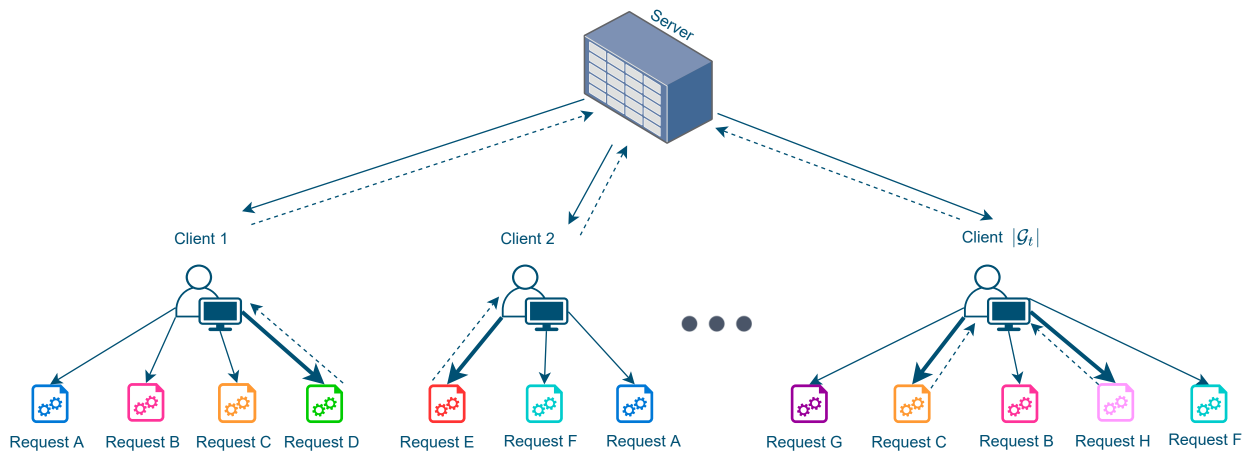

Figure 1: Illustration of a privacy-aware federated learning setup. A central server can offer different clients, who have varying privacy requirements, tasks that require the client to train a model using a given portion of their local data set.

certain privacy threshold ($\gamma_{t,G}$). The context coming at each round t for a base arm m, denoted by $x_{t,m}$, is the data set usage percentage for each client-request pair and hence affects base arm outcomes. The server cares about the clients' privacy requirements as privacy is an important factor which influences the likelihood of a client enrolling in future crowdsourcing jobs. Essentially, a client is less likely to participate in future crowdsourcing jobs from a server if the server repeatedly requests the client to perform tasks which cause data leakage beyond the privacy leakage preference of the client. Also in this setup, it is not possible for clients to reject tasks which are assigned to them, however, if the server assigns tasks which disregard the privacy preference of the clients, the clients may leave after finishing their tasks and never join for another crowdsourcing job with the same server which could be a serious problem.

3 Algorithm

We design an optimistic algorithm that uses GP UCBs in order to identify groups that satisfy thresholds and super arms that maximize rewards. We name our algorithm Thresholded Combinatorial GP-UCB (TCGP-UCB) with pseudo-code given in Algorithm 1. TCGP-UCB uses the *double UCB* principle, using different exploration bonuses for group and super arm selection. These bonuses are adjusted based on the trade-off parameter ζ given in the regret definition in equation 1.

For every available base arm, two indices which are upper confidence bounds on the two expected outcomes are defined by considering the parameter $\zeta \in [0, 1]$. Consider base arm m with its associated context $x_{t,m}$. We define its reward index as

$$i_t(x_{t,m}) = \mu_{1 [\![t-1]\!]}(x_{t,m}) + \frac{1}{1-\zeta}(\sqrt{\beta_{1,t}})\sigma_{1 [\![t-1]\!]}(x_{t,m}),$$

where $\mu_{1 [\![t-1]\!]}(x_{t,m})$ and $\sigma_{1 [\![t-1]\!]}(x_{t,m})^2$ stand for the posterior mean and variance. Similarly, we define the satisfying index of base arm m as

$$i'_t(x_{t,m}) = \mu_{2 [\![t-1]\!]}(x_{t,m}) + \frac{1}{\zeta}(\sqrt{\beta_{2,t}})\sigma_{2 [\![t-1]\!]}(x_{t,m}).$$

We give the indices $i'_t(x_{t,m})$ to $\text{Oracle}_{\text{grp}}$, which returns the groups whose expected rewards are greater than their respective thresholds among the feasible group set $\mathcal{G}_t$. We form the set of feasible super arms whose groups are expected to satisfy their thresholds, $\hat{\mathcal{S}}'_t$, by using the returned groups and $\mathcal{S}_t$. Namely, we have $\hat{\mathcal{S}}'_t = \{S \in \mathcal{S}_t : \forall(G \in \mathcal{G}_t), v_G(i'_t(\boldsymbol{x}_{t,G\cap S_t})) > \gamma_{t,G}\}$. Then, we give the indices $i_t(x_{t,m})$ of the base arms that construct $\hat{\mathcal{S}}'_t$ to $\text{Oracle}_{\text{spr}}$, which returns an α-optimal super arm.

Algorithm 1 TCGP-UCB

Input: $\mathcal{X}, K, M, \zeta$; GP Prior: $\mathcal{GP}(\mu_0, k)$
Initialize: μ_0, k
for $t = 1, \dots, T$ **do**
  Observe base arms in $\mathcal{M}_t$ and their contexts $\mathcal{X}_t$ as well as groups in $\mathcal{G}_t$
  for $x_{t,m} : m \in \mathcal{M}_t$ **do**
    Calculate $\boldsymbol{\mu}_{[\![t-1]\!]}(x_{t,m})$ and $\boldsymbol{\sigma}_{[\![t-1]\!]}(x_{t,m})$
  end for
  Compute indices $i_t(x_{t,m})$ and $i'_t(x_{t,m})$
  $\hat{\mathcal{G}}_t \leftarrow \text{Oracle}_{\text{grp}}(i'_t(x_{t,m})_{m\in\mathcal{M}_t}, \mathcal{G}_t)$
  Form the set $\hat{\mathcal{S}}'_t$
  $S_t \leftarrow \text{Oracle}_{\text{spr}}(i_t(x_{t,m})_{m\in\hat{\mathcal{S}}'_t}, \hat{\mathcal{S}}'_t)$
  Observe base arm outcomes in S_t, collect group rewards and super arm reward
end for

4 Theoretical Analysis

Next is our main result which asserts a high probability upper bound on the total regret of our algorithm in terms of $\overline{\gamma}_T$. The supplemental document contains further details of our results and this section contains analysis from (Nika et al., 2021) with additions of group regret and trade-off parameter. Throughout this section, we take $j \in \{1, 2\}$.

Theorem 1 *Super arm regret and group regret incurred by TCGP-UCB in T rounds are upper bounded with probability at least $1 - \delta_j$ where $\delta_j \in (0,1)$, $T \in \mathbb{N}$ and $\beta_{j,t} = 2\log(M\pi^2t^2/3\delta_j)$ as follows:*

$$R_g(T) \leq \sqrt{C_1\beta_{2,T}KT\overline{\gamma_2}_T},$$

where $C_1 = 2B^2(\frac{\zeta+1}{\zeta})^2(\lambda^* + \sigma^2)$ and

$$R_s(T) \leq \sqrt{C_2\beta_{1,T}KT\overline{\gamma_1}_T},$$

where $C_2 = 2{B'}^2(\frac{2-\zeta}{1-\zeta})^2(\lambda^* + \sigma^2)$. Hence, the total regret is bounded by:

$$R(T) \leq \sqrt{C\beta_T KT\overline{\gamma}_T},$$

where $C = 8(B + B')^2(\lambda^* + \sigma^2), \beta_T = \max\{\beta_{1,T}, \beta_{2,T}\}$ and $\overline{\gamma}_T = \max\{\overline{\gamma_1}_T, \overline{\gamma_2}_T\}$.

Next, we provide a regret bound that depends on the classical notion of maximum information gain. Here, we assume that $|S| = K$ for any $S \in \mathcal{S}$ where K is fixed.

Theorem 2 *Fix $\delta_j \in (0,1)$ and let $T, K \in \mathbb{N}$. Under the conditions of Theorem 1 and assuming that $|S| = K$ for any $S \in \mathcal{S}$, let $\overline{T} = \sum_{t=1}^{T}|S_t| = KT$. Then, the total regret incurred by TCGP-UCB in T rounds is upper bounded with the following with probability at least $1 - \delta_j$:*

$$R(T) \leq \sqrt{C\beta_T KT\gamma_{\overline{T}}},$$

where

$$\gamma_{\overline{T}} = \max_{A\subset\mathcal{X}:|A|=\overline{T}} I_j(\boldsymbol{r}_j(\boldsymbol{z}_A); \boldsymbol{f}_j(\boldsymbol{z}_A)) \tag{2}$$

while C and β_T are the same as in Theorem 1.

5 Experiments

We perform experiments using a privacy-aware federated learning setup to show that our algorithm outperforms the non-GP state-of-the-art while maintaining user privacy requirements. We perform further simulations in Appendix B to investigate the effect of the trade-off parameter, ζ.

5.1 Setup

We consider a privacy-aware federated learning setup comprised of a server and clients. The server's goal is to train a model using clients' data. However, clients do not wish to directly share their data with the server and instead train the model on a portion of their local data set, with the portion decided by the server, which they then send to the server. Thus, in each round, the server must serve requests to the available clients and select the portion of their data set to locally train on. Moreover, each client has different privacy requirements regarding the maximum amount of data that can be leaked.

In each round, we simulate the number of available clients using a Poisson distribution with mean 50. Then, the server can make a request to each client to ask them to use x portion of their data set for training the local model and then send the updated model to the server. We allow for $x \in \{0.01, 0.02, \ldots, 1.00\}$. Thus, for each client there are 100 possible requests. We then represent each request-client pair as a base arm with a one-dimensional context in $[0, 1]$ that represents how much of the data set the request asks for the client to use. We allow for the same client to receive multiple requests, as a client may have different local data sets on which he can train multiple local models. The maximum number of requests that each client can simultaneously receive in one round follow a Poisson distribution with mean 5.

After picking base arms (i.e., client-request pairs), the clients train the models on their local data sets and then return their model updates. We model the amount of useful information that the server gained from each request as a sigmoid-like function. More specifically, given context x, we have that the expected super arm outcome of x is $f_1(x) = \frac{1}{1+\exp(5-10x)}$. This model is justified by observing that very little data leads to overfitting and hence the first flat region for small x, and once the amount of data surpasses some threshold, noticeable gain in the amount of useful information transferable from the local model can be seen, hence the linear portion of f_1. Lastly, as the amount of data keeps increasing, we have diminishing returns in the amount of useful information, captured by the flat region of f_1 for large x. Then, the super arm reward is the sum of all the information learned from all requests. Thus, $u(f_1(\boldsymbol{x}_S)) = \sum_{i=1}^{|S|} f_1(x_i)$.

We model each client's privacy requirement using a privacy leakage threshold in $[0, 1]$, where clients intend for the amount of information leaked from their local data set to be below this threshold. When clients train on larger portions of their data sets, more of their training set and hence information is leaked (Melis et al., 2019). We model this leakage as $f_2(x) = 0.05 + 0.95 \exp(-5x)$. We justify this model by first noting that if very few training samples are used, then near-exact versions of the training samples can be extracted from the model updates or gradients (Melis et al., 2019). However, as the number of training samples increases, the leakage also decreases but never goes to 0. Finally, defining groups as set of client-request pairs from the same client, group reward–better called loss in this scenario–is the total leakage amount of all the requests of the client.[4]

5.2 Algorithms

We run the simulation using a slightly modified version of our algorithm, STCGP-UCB, and the non-GP C3-MAB state-of-the-art algorithm, ACC-UCB of (Nika et al., 2020).

[4]Even though group reward is defined as the non-leakage probability, considering it as the total leakage is also acceptable as our proofs still hold.

5.2.1 STCGP-UCB

We run a slightly modified version of our algorithm that uses sparse appromixation to GPs, called STCGP-UCB, where we use the sparse approximation to the GP posterior described in (Titsias, 2009). In this sparse approximation, instead of using all of the arm contexts up to round t to compute the posterior, a small s element subset of them is used, called the inducing points. By using a sparse approximation, the time complexity of the posterior updating procedure reduces from $O(K^3 + (2K)^3 + \ldots + (KT)^3) = O(K^3T^4)$ to $O(s^2KT^2)$. In our simulation, we set $s = 10$. We also set $\delta = 0.05$, $\zeta = 0.5$, and use two squared exponential kernels with both lengthscale and variance set to 1.

5.2.2 ACC-UCB

We set $v_1 = 1, v_2 = 1, \rho = 0.5$, and $N = 2$, as given in Definition 1 of (Nika et al., 2020). The initial (root) context cell, $X_{0,1}$, is a line centered at (0.5).

5.3 Results

We run the simulation for 100 rounds with eight independent runs, averaging over the results of each run. We plot the super arm and group regret in Figure 2. First, notice that ACC-UCB incurs linear group regret as it does not take group thresholds into account. On the other hand, STCGP-UCB incurs very low group regret as it accounts for group thresholds. Additionally, the super arm regret of STCGP-UCB is less than that of ACC-UCB, indicating that even though STCGP-UCB is balancing between minimizing both group and super arm regret, it still outperforms ACC-UCB. This is likely due to the use of GPs, which allow for faster convergence to $\boldsymbol{f}$.

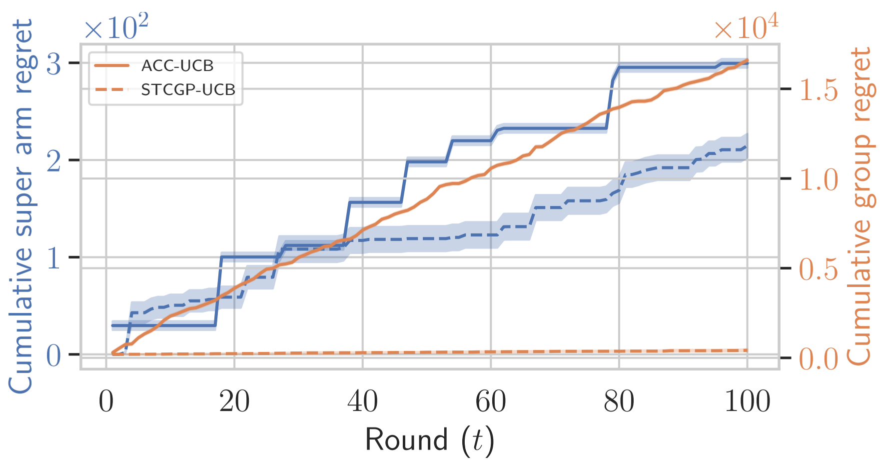

Figure 2: Cumulative super arm and group regret of ACC-UCB and STCGP-UCB. Super arm and group regret are represented by blue and orange, respectively. Moreover, ACC-UCB and STCGP-UCB are represented by solid and dashed lines, respectively. Shaded regions indicate ± std.

6 Conclusion

We considered the C3-MAB problem with semi-bandit feedback, where in each round the agent has to play a feasible subset of the base arms in order to maximize the cumulative reward while satisfy group constraints. Our setup was motivated by privacy aware FL, where the goal is to optimize overall training result while satisfying clients' privacy requirements. Under the assumption that the expected base arm outcomes are drawn from a two-output GP and that the expected reward is Lipschitz continuous with respect to the expected base arm outcomes, we proposed TCGP-UCB, a double UCB algorithm that incurs $\tilde{O}(\sqrt{KT\overline{\gamma}_T})$ regret in T rounds. In experiments, we showed that sparse GPs can be used to speed up UCB computation, while simultaneously outperforming the state-of-the-art non-GP-based C3-MAB algorithm. Our comparisons also indicated that GPs can transfer knowledge among contexts better than partitioning the contexts into groups of similar contexts based on a similarity metric. An interesting future research direction involves investigating how dependencies between base arms can be used for more efficient exploration. Then, when the oracle selects base arms sequentially, it is possible to update the posterior variances of the not yet selected base arms by conditioning on the selected, but not yet observed, base arms.

References

Nicolo Cesa-Bianchi and Gábor Lugosi. Combinatorial bandits. *J. Comput. Syst. Sci.*, 78(5):1404–1422, 2012.

Cangxiong Chen and Neill D. F. Campbell. Understanding training-data leakage from gradients in neural networks for image classification, 2021.

Lixing Chen, Jie Xu, and Zhuo Lu. Contextual combinatorial multi-armed bandits with volatile arms and submodular reward. In *Proc. Adv. Neural Inf. Process. Syst.*, pp. 3247–3256, 2018.

Wei Chen, Yajun Wang, and Yang Yuan. Combinatorial multi-armed bandit: General framework and applications. In *Proc. 30th Int. Conf. Mach. Learn.*, pp. 151–159. PMLR, 2013.

Wei Chen, Wei Hu, Fu Li, Jian Li, Yu Liu, and Pinyan Lu. Combinatorial multi-armed bandit with general reward functions. In *Proc. Adv. Neural Inf. Process. Syst.*, pp. 1659–1667, 2016a.

Wei Chen, Yajun Wang, Yang Yuan, and Qinshi Wang. Combinatorial multi-armed bandit and its extension to probabilistically triggered arms. *J. Mach. Learn. Res.*, 17(1):1746–1778, 2016b.

Andreas Krause and Cheng Ong. Contextual gaussian process bandit optimization. In J. Shawe-Taylor, R. Zemel, P. Bartlett, F. Pereira, and K. Q. Weinberger (eds.), *Adv. in Neural Inf. Process. Syst.*, volume 24. Curran Associates, Inc., 2011.

Lihong Li, Wei Chu, John Langford, and Robert E Schapire. A contextual-bandit approach to personalized news article recommendation. In *Proc. 19th Int. Conf. World Wide Web*, pp. 661–670, 2010.

Shuai Li, Baoxiang Wang, Shengyu Zhang, and Wei Chen. Contextual combinatorial cascading bandits. In *Proc. 33rd Int. Conf. Mach. Learn.*, pp. 1245–1253. PMLR, 2016.

Andrea Locatelli, Maurilio Gutzeit, and Alexandra Carpentier. An optimal algorithm for the thresholding bandit problem. In Maria Florina Balcan and Kilian Q. Weinberger (eds.), *Proc. of The 33rd Int. Conf. on Mach. Learn.*, volume 48, pp. 1690–1698. PMLR, 20–22 Jun 2016.

Tyler Lu, Dávid Pál, and Martin Pál. Contextual multi-armed bandits. In *Proc. 13th Int. Conf. Artif. Intell. and Statist.*, pp. 485–492, 2010.

Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-Efficient Learning of Deep Networks from Decentralized Data. In Aarti Singh and Jerry Zhu (eds.), *Proc. of the 20th Int. Conf. on Artif. Intell. and Statist.*, volume 54, pp. 1273–1282. PMLR, 20–22 Apr 2017.

Luca Melis, Congzheng Song, Emiliano De Cristofaro, and Vitaly Shmatikov. Exploiting unintended feature leakage in collaborative learning. In *2019 IEEE Symp. on Sec. and Priv. (SP)*, pp. 691–706. IEEE, 2019.

Subhojyoti Mukherjee, Kolar Purushothama Naveen, Nandan Sudarsanam, and Balaraman Ravindran. Thresholding bandits with augmented UCB. *CoRR*, abs/1704.02281, 2017.

Andi Nika, Sepehr Elahi, and Cem Tekin. Contextual combinatorial volatile multi-armed bandit with adaptive discretization. In *Proc. 23rd Int. Conf. Artif. Intell. and Statist.*, volume 108, pp. 1486–1496, 26–28 Aug 2020.

Andi Nika, Sepehr Elahi, and Cem Tekin. Contextual combinatorial bandits with changing action sets via gaussian processes. *CoRR*, abs/2110.02248, 2021. URL `https://arxiv.org/abs/2110.02248`.

Lijing Qin, Shouyuan Chen, and Xiaoyan Zhu. Contextual combinatorial bandit and its application on diversified online recommendation. In *Proc. of the 2014 SIAM Int. Conf. on Data Min.*, pp. 461–469. SIAM, 2014.

Paul Reverdy, Vaibhav Srivastava, and Naomi Ehrich Leonard. Satisficing in multi-armed bandit problems. *IEEE Trans. on Auto. Cont.*, 62(8):3788–3803, 2017. doi: 10.1109/TAC.2016.2644380.

Herbert Robbins. Some aspects of the sequential design of experiments. *Bulle. of the Amer. Math. Soc.*, 58 (5):527–535, 1952.

Shubhanshu Shekhar, Tara Javidi, et al. Gaussian process bandits with adaptive discretization. *Electron. J. Stat.*, 12(2):3829–3874, 2018.

Aleksandrs Slivkins. Contextual bandits with similarity information. In *Proc. 24th Annu. Conf. Learn. Theory.*, pp. 679–702, 2011.

N. Srinivas, A. Krause, S. M. Kakade, and M. W. Seeger. Information-theoretic regret bounds for gaussian process optimization in the bandit setting. *IEEE Trans. Inf. Theory*, 58(5):3250–3265, 2012.

Michalis K Titsias. Variational learning of inducing variables in sparse gaussian processes. In *Proc. 12th Int. Conf. Artif. Intell. and Statist.*, pp. 567–574, 2009.

Nguyen Truong, Kai Sun, Siyao Wang, Florian Guitton, and YiKe Guo. Privacy preservation in federated learning: An insightful survey from the gdpr perspective. *Comp. & Sec.*, 110:102402, 2021. ISSN 0167-4048. doi: https://doi.org/10.1016/j.cose.2021.102402.

Kang Wei, Jun Li, Ming Ding, Chuan Ma, Howard H. Yang, Farhad Farokhi, Shi Jin, Tony Q. S. Quek, and H. Vincent Poor. Federated learning with differential privacy: Algorithms and performance analysis. *IEEE Trans. on Inf. Foren. and Sec.*, 15:3454–3469, 2020. doi: 10.1109/TIFS.2020.2988575.

Christopher KI Williams and Francesco Vivarelli. Upper and lower bounds on the learning curve for gaussian processes. *Mach. Learn.*, 40(1):77–102, 2000.

Mengwei Yang, Linqi Song, Jie Xu, Congduan Li, and Guozhen Tan. The tradeoff between privacy and accuracy in anomaly detection using federated xgboost, 2019.