Robust Contextual Bandits via Bootstrapping

Qiao Tang, Hong Xie, Yunni Xia, Jia Lee, Qingsheng Zhu

Chongqing Key Laboratory of Software Theory and Technology, Chongqing University
felicity_0719@163.com, xiehong2018@cqu.edu.cn, xiayunni@hotmail.com, {lijia,qszhu}@cqu.edu.cn

Abstract

Upper confidence bound (UCB) based contextual bandit algorithms require one to know the tail property of the reward distribution. Unfortunately, such tail property is usually unknown or difficult to specify in real-world applications. Using a tail property heavier than the ground truth leads to a slow learning speed of the contextual bandit algorithm, while using a lighter one may cause the algorithm to diverge. To address this fundamental problem, we develop an estimator (evaluated from historical rewards) for the contextual bandit UCB based on the multiplier bootstrapping technique. We first establish sufficient conditions under which our estimator converges asymptotically to the ground truth of contextual bandit UCB. We further derive a second order correction for our estimator so as to obtain its confidence level with a finite number of rounds. To demonstrate the versatility of the estimator, we apply it to design a BootLinUCB algorithm for the contextual bandit. We prove that the BootLinUCB has a sub-linear regret upper bound and also conduct extensive experiments to validate its superior performance.

Introduction

Contextual bandit is a popular online learning framework and has been applied to solve many real-world problems, i.e., it has been applied to recommender systems to recommend products to interactive users (Li et al. 2010; Wang, Wu, and Wang 2016; Zhang et al. 2019), applied to optimize information retrieval algorithms (Hofmann et al. 2011; Gampa and Fujita 2019; Glowacka et al. 2019), as well as applied to networking applications such as selecting edge servers in mobile edge computing systems (Ouyang et al. 2019). Also, numerous variants of contextual bandit were developed (Filippi et al. 2010; Wang, Wu, and Wang 2016; Krishnamurthy, Wu, and Syrgkanis 2018; Zhang et al. 2019). To illustrate, consider the following example of a simplified contextual bandit:

Example 1. *Consider a finite number of arms indexed by $a \in \mathcal{A} \triangleq \{1, \ldots, A\}$, where $A \in \mathbb{N}_+$. Arm a is associated with a feature vector $\boldsymbol{x}_a \in \mathbb{R}^d$, where $d \in \mathbb{N}_+$. The reward model for arm a is*

$$R_a = \boldsymbol{x}_a^T \boldsymbol{\theta}_* + W_a,$$

where $\boldsymbol{\theta}_* \in \mathbb{R}^d$ and W_a is random variable with normal distribution $\mathcal{N}(0, \sigma_a^2)$. In each round $t \in \mathbb{N}_+$, the decision maker selects an arm $a_t \in \mathcal{A}_t \subseteq \mathcal{A}$ and receives the reward, which is a sample from R_{a_t}. The objective is to design an arm selection algorithm to attain as large as possible the cumulative reward in $T \in \mathbb{N}_+$ rounds.

For the bandit feedback illustrated in Example 1, contextual bandit algorithms need to balance the *exploitation vs. exploration* trade-off. One popular class of contextual bandit algorithms use the upper confidence bound (UCB) approach to balance this trade-off (Abbasi-Yadkori, Pál, and Szepesvári 2011; Auer 2002; Chu et al. 2011; Lattimore and Szepesvári 2020). The following example illustrates one of the UCB approaches for Example 1.

Example 2. *Consider the setting of Example 1. In round t, selects the arm $a_t \in \arg\max_{a \in \mathcal{A}_t} U_t(a)$, where $U_t(a)$ is the UCB associated with arm a and it is defined as*

$$U_t(a) \triangleq \boldsymbol{x}_a^T \widehat{\boldsymbol{\theta}}_t + \phi_t(\boldsymbol{x}_a, \boldsymbol{\sigma}, \mathcal{H}_{t-1}),$$

where $\boldsymbol{\sigma} = (\sigma_1, \ldots, \sigma_A)$, $\mathcal{H}_t$ denotes the historical arms and rewards up to round t, $\widehat{\boldsymbol{\theta}}_t$ denote an estimator of $\boldsymbol{\theta}_*$ evaluated from $\mathcal{H}_{t-1}$, and the penalty term satisfies $\phi_t(\boldsymbol{x}_a, \boldsymbol{\sigma}, \mathcal{H}_{t-1}) > 0$.

Example 2 summarizes the structure of LinUCB algorithms for contextual bandit (Abbasi-Yadkori, Pál, and Szepesvári 2011; Chu et al. 2011). The penalty term encourages exploration and a larger value induces more exploration. The penalty term $\phi_t(\boldsymbol{x}_a, \boldsymbol{\sigma}, \mathcal{H}_{t-1})$ is non-decreasing in $\sigma_a, \forall a \in \mathcal{A}$, capturing that when the reward is subjected to a larger variation, the estimated $\widehat{\boldsymbol{\theta}}_t$ becomes less accurate, thus, the algorithm needs to do more explorations. Unfortunately, one does not possess the knowledge of $\sigma_a, \forall a \in \mathcal{A}$, in practice. Using a standard deviation larger than σ_a for Algorithm in Example 2 leads to over exploration, i.e., slow learning speed, while the reverse may cause the algorithm to diverge. In summary, Example 2 highlights a reward distribution tail property mis-match problem (e.g., σ_a characterizes the tail property of the reward distribution), which is inherent in many UCB based contextual bandit algorithms beyond LinUCB, e.g., LinRel (Auer 2002), SupLinUCB (Chu et al. 2011), action elimination algorithms (Lattimore and Szepesvári 2020).

This paper considers how to address the fundamental problem of reward distribution tail property mis-match in a general setting, i.e., the reward distribution can evolve over time and can have non-parametric distribution beyond Gaussian, etc. Specifically, we deploy multiplier bootstrap methods (Arlot et al. 2010; Yang, Shang, and Cheng 2017) and develop an estimator for the UCB of contextual bandit, which can be directly evaluated from the historical arms and rewards without requiring one to know or specify the tail property of the reward distribution. There are two challenges in providing theoretical guarantees: *(1) the ground truth UCB being estimated across arms are correlated, (2) one only possesses the bandit feedback in each round and it is coupled with the estimated UCB.* We address them and our contributions are:

- We apply the multiplier bootstrap technique to develop a novel estimator for the contextual bandit UCB. The estimator can be directly evaluated from the historical arms and rewards without requiring one to specify the tail property of the reward distribution.
- We establish sufficient conditions, under which our estimator converges asymptotically to the ground truth of the contextual bandit UCB. These sufficient conditions guide us to select arms in early time decision rounds.
- We further derive a second-order correction for our estimator to obtain its confidence level with only a finite number of rounds. We select arm associated with the largest data-driven UCB (i.e., corrected estimator) in the remaining time slots (i.e., except the early time slots mentioned above), resulting in our BootLinUCB algorithm for the contextual bandit. We prove BootLinUCB has a sub-linear regret upper bound. The estimator is general and can be applied to other UCB based contextual bandits, e.g., LinRel (Auer 2002), SupLinUCB (Chu et al. 2011), etc.
- We conduct extensive experiments to validate its superior performance of our BootLinUCB algorithm over the latest bootstrapping based LinUCB algorithm (Hao et al. 2019) and the classical LinUCB algorithm (Chu et al. 2011).

Related Work

The research of contextual bandit (Chu et al. 2011; Li et al. 2010) can be organized into three lines: *algorithmic line, modeling line* and *application line.* The algorithmic line focuses on developing algorithms with faster learning speed, more robust to model misspecification, etc., (Abbasi-Yadkori, Pál, and Szepesvári 2011; Chu et al. 2011; Agrawal and Goyal 2013; Zhu et al. 2017; Hao et al. 2019). The modeling line focuses on extending the contextual bandit model to handle more general or complicated settings (Filippi et al. 2010; Wang, Wu, and Wang 2016; Krishnamurthy, Wu, and Syrgkanis 2018; Zhang et al. 2019). The application line focuses on tuning contextual bandit algorithms to solve online decision problems in real-world applications (Li et al. 2010; Hofmann et al. 2011; Glowacka et al. 2019; Ouyang et al. 2019; Zhang et al. 2019). This paper falls into the algorithmic line, in particular, using bootstrapping methods to improve contextual bandit algorithms.

A number of recent works applied bootstrapping methods to design data driven algorithms for contextual bandit. These research can be organized into the Thompson sampling line and UCB line. In the Thompson sampling line, bootstrapping methods are used to design Thompson sampling algorithms for contextual bandit (Eckles and Kaptein 2014; Osband and Van Roy 2015; Tang et al. 2015; Elmachtoub et al. 2017; Kveton et al. 2019; Vaswani et al. 2018). One advantage of the Thompson sampling method is that these algorithms are non-parametric, i.e., they do not require the parametric form on the model. However, these algorithms usually do not have theoretical guarantee, i.e., the regret upper bound, except (Kveton et al. 2019) whose regret upper bound is obtained under Bernoulli assumption. In contrast, our work applies to a broader class of distributions, i.e., symmetry sub-Gaussian distributions. Another technical difference is that our work develops data driven UCB algorithms for contextual bandit with theoretical guarantees. In the UCB line, bootstrapping methods are used to design data driven UCB algorithms for contextual bandit (Sudarsanam and Ravindran 2016; Hao et al. 2019). Again, one advantage over the UCB method is that these algorithms are non-parametric and adaptively adjusted to the ground truth UCB. These algorithms (Sudarsanam and Ravindran 2016; Hao et al. 2019) do not have regret upper bound for contextual bandit problem. Note that the algorithm in (Hao et al. 2019) provides regret upper bound for classical multi-armed bandit problem. However, it is non-trivial to extend it to contextual bandit problems because in the classical multi-armed bandit problem, the UCB for each arm is independent, while in contextual bandit they are correlated. We develop apply the multiplier bootstrap technique to develop a novel estimator for the contextual bandit UCB and establish conditions to guarantee the convergence of the estimator.

Contextual Bandit Model

Consider a contextual bandit model with a finite number of $A \in \mathbb{N}_+$ arms. Denote the arm set as $\mathcal{A} \triangleq \{1, \dots, A\}$. Each arm $a \in \mathcal{A}$ is associated with a $d \in \mathbb{N}_+$ dimensional feature vector $\boldsymbol{x}_a \in \mathbb{R}^d$. The feature vector $\boldsymbol{x}_a$ is known to the decision maker. Consider a discrete time system indexed by $t \in \mathbb{N}_+$. Each time slot corresponds to one decision epoch or decision round. In time slot t, a subset of $\mathcal{A}_t \subseteq \mathcal{A}$ arms is presented to the decision maker. Then, the decision maker selects one arm from $\mathcal{A}_t$ denoted by $a_t \in \mathcal{A}_t$. Finally, the decision maker receives a reward $r_t \in \mathbb{R}$. Note that the reward of other arms are not revealed to the decision maker. The objective is to design an arm selection algorithm to attain as high as possible the cumulative reward.

We consider linear reward functions. Define the reward function for arm a in time slot t as

$$R_{a,t} \triangleq \boldsymbol{x}_a^T \boldsymbol{\theta}_* + W_{a,t}, \tag{1}$$

where $\boldsymbol{\theta}_*$ denotes a "preference" vector and the random variable $W_{a,t}$ denotes a stochastic noise with zero mean $\mathbb{E}[W_{a,t}] = 0$. Furthermore, $W_{a,t}$ across arms and time slots are independent (Abbasi-Yadkori, Pál, and Szepesvári 2011; Chu et al. 2011). The preference vector $\boldsymbol{\theta}_*$ is unknown to the decision maker. The stochastic noise $W_{a,t}$ captures the randomness or variation in reward. The reward r_t in time slot t is a sample or realization of $R_{a_t,t}$. Then the expected reward in time slot t is $\mathbb{E}[R_{a_t,t}] = \boldsymbol{x}_{a_t}^T \boldsymbol{\theta}_*$.

We consider a risk neutral decision maker, who aims to maximize the expected cumulative reward $\mathbb{E}[\sum_{t=1}^T R_{a_t,t}]$. The linearity of expectation

$$\mathbb{E}\left[\sum_{t=1}^T R_{a_t,t}\right] = \sum_{t=1}^T \mathbb{E}[R_{a_t,t}]$$

implies that the optimal arm denoted by a_t^* in time slot t can be derived as

$$a_t^* \in \arg\max_{a\in\mathcal{A}_t} \mathbb{E}[R_{a,t}] = \arg\max_{a\in\mathcal{A}_t} \boldsymbol{x}_a^T \boldsymbol{\theta}_*.$$

However, the optimal arm a_t^* is unknown to the decision maker because the preference vector $\boldsymbol{\theta}_*$ is unknown to the decision maker. The objective is to design an arm selection algorithm, denoted by $\mathbb{A}$, to maximize the expected cumulative reward $\mathbb{E}[\sum_{t=1}^T R_{a_t,t}]$.

We consider a class of history-dependent arm selection algorithms $\mathbb{A}$, which prescribe an arm for each interaction history. Denote the reward history up to time slot t as $\mathcal{H}_t \triangleq \{(a_1, \boldsymbol{x}_{a_1}, r_1), \dots, (a_t, \boldsymbol{x}_{a_t}, r_t)\}$, which contains historical arms and rewards. Formally, the algorithm can be represented as a mapping function $\mathbb{A} : \mathcal{H}_{t-1} \mapsto \mathcal{A}_t$ and $a_t = \mathbb{A}(\mathcal{H}_{t-1})$. We use the regret to quantify the performance of algorithm $\mathbb{A}$, which is:

$$\mathcal{R}_T(\mathbb{A}) \triangleq \sum_{t=1}^T \boldsymbol{x}_{a_t^*}^T \boldsymbol{\theta}_* - \mathbb{E}\left[\sum_{t=1}^T \boldsymbol{x}_{a_t}^T \boldsymbol{\theta}_* \middle| a_t = \mathbb{A}(\mathcal{H}_{t-1})\right].$$

A smaller regret implies that algorithm $\mathbb{A}$ achieves a larger expected cumulative reward.

BootLinUCB Algorithmic Framework

In this section, we first present some basic elements of regularized least squares for contextual bandits. Then we present the formulation of our *quantile bootstrapping oracle* for contextual bandits. Finally, we apply this quantile bootstrapping oracle to design our BootLinUCB algorithmic framework.

Regularized Least Squares

Regularized least squares is the main stream method to estimate the preference vector $\boldsymbol{\theta}_*$ of contextual bandits (Lattimore and Szepesvári 2020):

$$\widehat{\boldsymbol{\theta}}_t = \arg\min_{\boldsymbol{\theta}\in\mathbb{R}^d} \sum_{s=1}^{t-1} (\boldsymbol{x}_{a_s}^T \boldsymbol{\theta} - r_s)^2 + \lambda\|\boldsymbol{\theta}\|_2^2,$$

where $\widehat{\boldsymbol{\theta}}_t$ denotes an estimation of $\boldsymbol{\theta}_*$ in time slot t and $\lambda > 0$ denotes a regularization parameter. The closed form expression for $\widehat{\boldsymbol{\theta}}_t$ is:

$$\widehat{\boldsymbol{\theta}}_t = \boldsymbol{V}_{t-1}^{-1} \sum_{s=1}^{t-1} \boldsymbol{x}_{a_s} r_s,$$

where

$$\boldsymbol{V}_{t-1} = \lambda \boldsymbol{I} + \sum_{s=1}^{t-1} \boldsymbol{x}_{a_s} \boldsymbol{x}_{a_s}^T.$$

This closed form expression of $\widehat{\boldsymbol{\theta}}_t$ implies that

$$\boldsymbol{x}_a^T(\boldsymbol{\theta}_* - \widehat{\boldsymbol{\theta}}_t) = E(\boldsymbol{x}_a, \mathcal{F}_{t-1}) + \lambda \boldsymbol{x}_a^T \boldsymbol{V}_{t-1}^{-1} \boldsymbol{\theta}_*,$$

where $\mathcal{F}_{t-1} \triangleq \{a_1, \dots, a_{t-1}\}$, and $E(\boldsymbol{x}_a, \mathcal{F}_{t-1})$ is defined as a partial residual

$$E(\boldsymbol{x}_a, \mathcal{F}_{t-1}) \triangleq \boldsymbol{x}_a^T \boldsymbol{V}_{t-1}^{-1} \sum_{s=1}^{t-1} \boldsymbol{x}_{a_s} (\boldsymbol{x}_{a_s}^T \boldsymbol{\theta}_* - r_s).$$

The tail property of the reward distribution is essential for deriving the UCB of $\boldsymbol{x}_a^T \boldsymbol{\theta}_*$. To illustrate, the following lemma generalizes the UCB in (Chu et al. 2011) for a bounded reward $[0, 1]$ to a sub-Gaussian reward.

Lemma 1. *Suppose $W_{a,t}$ is sub-Gaussian, i.e., $\mathbb{E}[\exp(cW_{a,t})] \leq \exp(c^2\sigma_{a,t}^2/2), \forall c \in \mathbb{R}$, and $\mathcal{F}_t$ is deterministic, then*

$$\mathbb{P}[\boldsymbol{x}_a^T \boldsymbol{\theta}_* \geq \boldsymbol{x}_a^T \widehat{\boldsymbol{\theta}}_t + \lambda \boldsymbol{x}_a^T \boldsymbol{V}_{t-1}^{-1} \boldsymbol{\theta}_* + \varphi_t(\boldsymbol{x}_a, \mathcal{H}_{t-1})] \leq \alpha_t,$$

where

$$\varphi_t(\boldsymbol{x}_a, \mathcal{H}_{t-1}) = \sigma_{\max} \sqrt{2\ln(1/\alpha_t)} \|\boldsymbol{x}_a\|_{\boldsymbol{V}_{t-1}^{-1}}$$

denotes an upper bound of the $(1-\alpha_t)$-quantile of the partial residual $E(\boldsymbol{x}_a, \mathcal{F}_{t-1})$ and $\sigma_{\max} = \max_{a,t} \sigma_{a,t}$.

Remark: Lemma 1 implies that an UCB for the ground truth reward $\boldsymbol{x}_a^T \boldsymbol{\theta}_*$ can be

$$U_t(a) \triangleq \boldsymbol{x}_a^T \widehat{\boldsymbol{\theta}}_t + \lambda \boldsymbol{x}_a^T \boldsymbol{V}_{t-1}^{-1} \boldsymbol{\theta}_* + \varphi_t(\boldsymbol{x}_a, \mathcal{H}_{t-1}).$$

A variety of contextual bandit algorithms use $U_t(a)$ to select arms: LinUCB algorithm (Chu et al. 2011) selects arms via $a_t \in \arg\max_{a\in\mathcal{A}_t} U_t(a)$; LinRel (Auer 2002) and SupLinUCB (Chu et al. 2011) use $U_t(a)$ to assists arm selection and sub-optimal arm elimination; forced-exploration based algorithms (Abbasi-Yadkori, Antos, and Szepesvári 2009) can use to $U_t(a)$ to determine the condition of stopping exploration adaptively, just to name a few. However, in practice, the exact $\varphi_t(\boldsymbol{x}_a, \mathcal{H}_{t-1})$ is unknown to the decision maker, because the parameter $\sigma_{a,t}$ is unknown and $W_{a,t}$ may not even be sub-Gaussian, making it more difficult to specify the UCB. We next formulate an oracle to bootstrap $U_t(a)$ beyond sub-Gaussian reward.

Quantile Bootstrapping Oracle

To be consistent with the condition in Lemma 1, in this subsection, we consider a deterministic $\mathcal{F}_t$. Lemma 1 shows that the upper confidence bound of the ground truth reward $\boldsymbol{x}_a^T \boldsymbol{\theta}_*$ is determined by the $(1 - \alpha_t)$-quantile of the partial residual $E(\boldsymbol{x}_a, \mathcal{F}_{t-1})$. Formally, the ground truth $(1 - \alpha)$-quantile of $E(\boldsymbol{x}_a, \mathcal{F}_{t-1})$ is defined as

$$q_t(\boldsymbol{x}_a, 1-\alpha) \triangleq \inf\left\{z\in\mathbb{R} \middle| \mathbb{P}\left[E(\boldsymbol{x}_a, \mathcal{F}_{t-1}) \leq z\right] \geq 1-\alpha\right\}.$$

With the ground truth quantile $q_t(\boldsymbol{x}_a, 1 - \alpha_t)$, the ground truth UCB for $\boldsymbol{x}_a^T \boldsymbol{\theta}_*$ can be expressed as

$$\Upsilon_t(a) \triangleq \boldsymbol{x}_a^T \widehat{\boldsymbol{\theta}}_t + \lambda \boldsymbol{x}_a^T \boldsymbol{V}_{t-1}^{-1} \boldsymbol{\theta}_* + q_t(\boldsymbol{x}_a, 1 - \alpha_t).$$

Note that $\Upsilon_t(a) \leq U_t(a)$ because $\varphi_t(\boldsymbol{x}_a, \mathcal{H}_{t-1})$ is an upper bound of the $(1 - \alpha_t)$-quantile of the partial residual $E(\boldsymbol{x}_a, \mathcal{F}_{t-1})$, i.e., $q_t(\boldsymbol{x}_a, 1 - \alpha_t) \leq \varphi_t(\boldsymbol{x}_a, \mathcal{H}_{t-1})$. The following definition states a class of bootstrapping oracles, which bootstrap (or estimate) the quantile $q_t(\boldsymbol{x}_a, 1-\alpha_t)$ from the reward history $\mathcal{H}_{t-1}$ with theoretical guarantees.

Definition 1. *Define* $\texttt{BootQuantile}(\boldsymbol{x}, \mathcal{H}_{t-1}, \alpha)$ *as an oracle which bootstraps the quantile* $q_t(\boldsymbol{x}_a, \alpha)$ *from the interaction history* $\mathcal{H}_{t-1}$*, and satisfies:*

$$\mathbb{P}\left[E(\boldsymbol{x}_a, \mathcal{F}_{t-1}) \leq Q_t(\boldsymbol{x}_a, \alpha)\right] \geq \alpha, \forall a \in \mathcal{A}_t,$$

where $Q_t(\boldsymbol{x}_a, \alpha) = \texttt{BootQuantile}(\boldsymbol{x}_a, \mathcal{H}_{t-1}, \alpha)$, $\alpha \in [0, 1]$*, and* $\mathcal{F}_{t-1}$ *is deterministic.*

The oracle defined in Definition 1 takes the feature vector, interaction history and confidence level as input, and outputs an estimated quantile for each arm satisfying the inputted confidence level. The detail of the bootstrapping oracle is deferred to next section. In this section, let us focus on how to use it to design algorithms for contextual bandit.

BootLinUCB Algorithm

We apply the $\texttt{BootQuantile}(\boldsymbol{x}, \mathcal{H}_{t-1}, \alpha)$ oracle to design our BootLinUCB algorithmic framework outlined in Algorithm 1, where $\mathcal{F}_t$ can be coupled with reward. Note that for other algorithms like forced-exploration based algorithms (Abbasi-Yadkori, Antos, and Szepesvári 2009), LinRel (Auer 2002), SupLinUCB (Chu et al. 2011), where $\mathcal{F}_t$ is deterministic or independent of the reward, the $\texttt{BootQuantile}(\boldsymbol{x}, \mathcal{H}_{t-1}, \alpha)$ oracle can also be applied. Due to page limit, we omit them. To execute algorithm 1, one needs to specify a bootstrapping oracle $\texttt{BootQuantile}(\boldsymbol{x}_a, \mathcal{H}_{t-1}, \alpha)$ and the parameters for the confidence level $\alpha_t, \forall t = 1, \dots, T$. In each time slot t, the algorithm first computes an estimate of the preference parameter denoted by $\widehat{\boldsymbol{\theta}}_t$. It then computes an estimate of the quantile $Q_t(\boldsymbol{x}_a, 1 - \alpha_t)$ for each arm by calling the bootstrapping oracle. It uses the estimated quantile to construct a UCB for each arm, and then selects the arm with the largest UCB value. Finally, it receives the reward and updates the interaction history, etc. Note that $\lambda \boldsymbol{x}_a^T \boldsymbol{V}_{t-1}^{-1} \boldsymbol{\theta}_*$ is unknown as $\boldsymbol{\theta}_*$ is unknown. In the implementation, one can replace $\lambda \boldsymbol{x}_a^T \boldsymbol{V}_{t-1}^{-1} \boldsymbol{\theta}_*$ with its upper bound (Chu et al. 2011) $\lambda \boldsymbol{x}_a^T \boldsymbol{V}_{t-1}^{-1} \boldsymbol{\theta}_* \leq L\sqrt{\lambda} \|\boldsymbol{x}_a\|_{\boldsymbol{V}_{t-1}^{-1}}$, where L is an upper bound of the norm of the preference parameter $\|\boldsymbol{\theta}_*\| \leq L$. The following theorem states the regret upper bound.

Algorithm 1 BootLinUCB algorithmic framework

```
Input: an oracle $\texttt{BootQuantile}(\boldsymbol{x}_a, \mathcal{H}_{t-1}, \alpha)$ and
  confidence level parameters $\alpha_1, \dots, \alpha_T$.
$\mathcal{H}_0 \leftarrow \emptyset$, $\boldsymbol{b}_0 \leftarrow \boldsymbol{0}$, $\boldsymbol{V}_0 \leftarrow \lambda \boldsymbol{I}$.
for $t = 1, \dots, T$ do
    $\widehat{\boldsymbol{\theta}}_t = \boldsymbol{V}_{t-1}^{-1} \boldsymbol{b}_{t-1}$.
    $Q_t(\boldsymbol{x}_a, 1-\alpha_t) \leftarrow \texttt{BootQuantile}(\boldsymbol{x}_a, \mathcal{H}_{t-1}, 1-\alpha_t)$.
    Choose arm
        $a_t \in \arg\max_{a \in \mathcal{A}_t} \left[\boldsymbol{x}_a^T \widehat{\boldsymbol{\theta}}_t + \lambda \boldsymbol{x}_a^T \boldsymbol{V}_{t-1}^{-1} \boldsymbol{\theta}_* + Q_t(\boldsymbol{x}_a, 1-\alpha_t)\right]$.
    Observe reward $r_t$.
    $\boldsymbol{V}_t \leftarrow \boldsymbol{V}_{t-1} + \boldsymbol{x}_{a_t} \boldsymbol{x}_{a_t}^T$.
    $\boldsymbol{b}_t \leftarrow \boldsymbol{b}_{t-1} + \boldsymbol{x}_{a_t} r_t$.
    $\mathcal{H}_t \leftarrow \mathcal{H}_{t-1} \cup \{(a_t, \boldsymbol{x}_{a_t}, r_t)\}$.
end for
```

Theorem 1. *If for each given* a, $W_{a,t}, \forall t$ *are identical, the regret of Algorithm 1 can be bounded as*

$$\begin{aligned}
&\mathcal{R}_T(\mathbb{A}_{\texttt{BootLinUCB}}) \\
&\leq \sum_{t=1}^{T} \mathbb{E}\left[\min\{Q_t(\boldsymbol{x}_{a_t}, 1-\alpha_t) + Q_t(-\boldsymbol{x}_{a_t}, 1-\alpha_t), g_t\}\right] \\
&\quad + 2\sum_{t=1}^{T} \binom{t+A-2}{A-1} \alpha_t g_t,
\end{aligned}$$

where $g_t \triangleq \max_{a \in \mathcal{A}_t} \boldsymbol{x}_a^T \boldsymbol{\theta}_* - \min_{a \in \mathcal{A}_t} \boldsymbol{x}_a^T \boldsymbol{\theta}_*$ *denotes the maximum possible regret in time slot* t.

Remark: *Due to page limit, we present all proofs in our supplementary file.* Though in Algorithm 1 $\mathcal{F}_t$ depends on the reward, we can decouple it via the conditioning trick in the analysis. The term $\sum_{t=1}^{T} \binom{t+A-2}{A-1} \alpha_t g_t$ has an order of $O(\sum_{t=1}^{T} \binom{t+A-2}{A-1} \alpha_t)$. For example, if we select $\alpha_t = 1/(\binom{t+A-2}{A-1} t)$, then we have $O(\sum_{t=1}^{T} \binom{t+A-2}{A-1} \alpha_t) = O(\ln T)$. Namely, this term can be made sub-linear. To analyze the order of $\sum_{t=1}^{T} \mathbb{E}\left[\min\{Q_t(\boldsymbol{x}_{a_t}, 1-\alpha_t) + Q_t(-\boldsymbol{x}_{a_t}, 1-\alpha_t), g_t\}\right]$ we need more details of the bootstrapping oracle, and we defer it to next section.

Bootstrapping Algorithm

We first apply the multiplier bootstrap technique to design an estimator for the quantile $q_t(\boldsymbol{x}_a, \alpha)$. We establish sufficient conditions, under which our estimator converges. These conditions and the estimator guide us to design an algorithm to implement the bootstrapping oracle.

Asymptotically Accurate Estimator

To be consistent with the condition in definition 1, in this subsection, we consider a deterministic $\mathcal{F}_t$. The quantile $q_t(\boldsymbol{x}_a, \alpha)$ can be rewritten as

$$\begin{aligned}
&q_t(\boldsymbol{x}_a, \alpha) \\
&= \inf\left\{ z \in \mathbb{R} \,\middle|\, \mathbb{P}\left[\boldsymbol{x}_a^T \boldsymbol{V}_{t-1}^{-1} \sum_{s=1}^{t-1} \boldsymbol{x}_{a_s}(r_s - \boldsymbol{x}_{a_s}^T \boldsymbol{\theta}_*) \leq z\right] \geq \alpha \right\}.
\end{aligned}$$

In the above quantile, the randomness is caused by reward $r_s, s = 1, \dots, t$, which are independent samples from the reward distribution expressed in Equation (1). We apply the multiplier bootstrapping technique (Arlot et al. 2010) to resample the reward $r_s, s = 1, \dots, t$, for the purpose of estimating the quantile $q_t(\boldsymbol{x}_a, \alpha)$. Formally, we define an estimator for $q_t(\boldsymbol{x}_a, \alpha)$ as:

$$\begin{aligned}
&\widehat{q}_t(\boldsymbol{x}_a, \boldsymbol{\epsilon}, \alpha) \\
&\triangleq \inf\left\{ z \in \mathbb{R} \,\middle|\, \mathbb{P}_{\boldsymbol{w}}\left[\sum_{s=1}^{t-1} w_s \boldsymbol{x}_a^T \boldsymbol{V}_{t-1}^{-1} \boldsymbol{x}_{a_s} \epsilon_s \leq z\right] \geq \alpha \right\},
\end{aligned}$$

where $w_1, \dots, w_{t-1}$ are independent and identically distributed (IID) random variables following the Rademacher distribution, i.e., $w_s = 1$ with probability 0.5 and $w_s = -1$ with probability 0.5. We define $\boldsymbol{w} \triangleq (w_1, \dots, w_{t-1})$, $\epsilon_s \triangleq$

$\boldsymbol{x}_{a_s}^T \boldsymbol{\theta}_* - r_s$ as reward residual, $\boldsymbol{\epsilon} \triangleq (\epsilon_1, \ldots, \epsilon_{t-1})$, and $\mathbb{P}_{\boldsymbol{w}}$ as computing probability with respect to randomness caused by $\boldsymbol{w}$.

To analyze the estimator $\widehat{q}_t(\boldsymbol{x}_a, \boldsymbol{\epsilon}, \alpha)$, we need to characterize the tail of the reward distribution. Let $\mathcal{Z}$ denote the space of random variables such that $W_{a,t} \in \mathcal{Z}, \forall a, t$. The convergence of the estimator $\widehat{q}_t(\boldsymbol{x}_a, \boldsymbol{\epsilon}, \alpha)$ relies on the tail property of $\mathcal{Z}$ defined as follows.

Definition 2. *Define a metric*

$$H_{\mathcal{Z}}(z) \triangleq \sup_{Z \in \mathcal{Z}} \frac{\mathbb{E}\left[Z^2; Z^2 > z\right]}{\mathbb{E}\left[Z^2\right]}$$

to quantify how heaviness of the tail of a space of random variables $\mathcal{Z}$, where $z \in \mathbb{R}_+$.

For each given z, a larger $H_{\mathcal{Z}}(z)$ implies that the tail of the space of random variables $\mathcal{Z}$ is heavier. The following assumption captures a class of distributions with "nice" tails.

Assumption 1. *The space of random variables $\mathcal{Z}$ satisfies*

$$\lim_{z \to \infty} H_{\mathcal{Z}}(z) \to 0.$$

Assumption 1 characterizes a broad class of distributions. For example, if $\mathcal{Z}$ is a space of random variables with a bounded domain, then Condition 1 holds. If $\mathcal{Z}$ is a space of sub-Gaussian random variables, then Assumption 1 also holds.

Assumption 2. *There exist $0 < c_1 < c_2$ such that the variance $Var(W_{a,t}) \in [c_1, c_2], \forall a, t$.*

The next condition is essential for the convergence of the quantile estimator $\widehat{q}_t(\boldsymbol{x}_a, \boldsymbol{\epsilon}, \alpha)$.

Condition 1. *For any $a \in \mathcal{A}$, it holds that*

$$\lim_{t \to \infty} \|\boldsymbol{x}_a\|_{\boldsymbol{V}_t^{-1}} \to 0.$$

Condition 1 identifies a sequence of well behaved actions. Due to correlation among the quantile of arms, the quantile estimator $\widehat{q}_t(\boldsymbol{x}_a, \boldsymbol{\epsilon}, \alpha)$ may not converge under poorly behaved action sequences. We will show how to design arm selection strategies to guarantee Condition 1 in next section.

Theorem 2. *Under Assumption 1, 2 and Condition 1, the quantile estimator $\widehat{q}_t(\boldsymbol{x}_a, \boldsymbol{\epsilon}, \alpha)$ converges to the ground truth $q_t(\boldsymbol{x}_a, \alpha)$, i.e.,*

$$\lim_{t \to \infty} |\widehat{q}_t(\boldsymbol{x}_a, \boldsymbol{\epsilon}, \alpha) - q_t(\boldsymbol{x}_a, \alpha)| = 0, \forall a, \alpha,$$

where $\mathcal{F}_{t-1}$ is deterministic.

Remark: Theorem 2 states the sufficient conditions under which the estimator $\widehat{q}_t(\boldsymbol{x}_a, \boldsymbol{\epsilon}, \alpha)$ converges to the ground truth. Namely, the estimator $\widehat{q}_t(\boldsymbol{x}_a, \boldsymbol{\epsilon}, \alpha)$ is asymptotically accurate. However, it is difficult to implement, as it requires the preference parameter $\boldsymbol{\theta}_*$. To relieve this difficulty, we replace $\boldsymbol{\theta}_*$ with $\widehat{\boldsymbol{\theta}}_t$. Formally, we express the new estimator as

$$\begin{aligned} &\widehat{q}_t(\boldsymbol{x}_a, \widehat{\boldsymbol{\epsilon}}, \alpha) \\ &\quad = \inf\left\{ z \in \mathbb{R} \,\middle|\, \mathbb{P}_{\boldsymbol{w}}\left[\sum_{s=1}^{t-1} w_s \boldsymbol{x}_a^T \boldsymbol{V}_{t-1}^{-1} \boldsymbol{x}_{a_s} \widehat{\epsilon}_s \leq z\right] \geq \alpha \right\}, \end{aligned} \quad (2)$$

where $\widehat{\epsilon}_s = \boldsymbol{x}_{a_s}^T \widehat{\boldsymbol{\theta}}_t - r_s$ denotes the empirical residual and $\widehat{\boldsymbol{\epsilon}} = (\widehat{\epsilon}_1, \ldots, \widehat{\epsilon}_{t-1})$. The following theorem establish the asymptotic convergence of this estimator.

Theorem 3. *Under the same conditions as Theorem 2, the quantile estimator $\widehat{q}_t(\boldsymbol{x}_a, \boldsymbol{\epsilon}, \alpha)$ converges to the ground truth $q_t(\boldsymbol{x}_a, \alpha)$, i.e.,*

$$\lim_{t \to \infty} |\widehat{q}_t(\boldsymbol{x}_a, \widehat{\boldsymbol{\epsilon}}, \alpha) - q_t(\boldsymbol{x}_a, \alpha)| = 0, \forall a, \alpha,$$

where $\mathcal{F}_{t-1}$ is deterministic.

Remark: Theorem 3 states sufficient conditions under which the quantile estimator $\widehat{q}_t(\boldsymbol{x}_a, \widehat{\boldsymbol{\epsilon}}, \alpha)$ converges to the ground truth quantile $q_t(\boldsymbol{x}_a, \alpha)$. Note that it requires the same condition as Theorem 2, i.e., no extra conditions are needed.

Non-Asymptotic Validity of Estimator

The quantile estimator expressed in Eq. (2) has the nice asymptotic accurate property. However, it can not be directly used to design our bootstrapping oracle as we do not known the confidence level of it. Now, we establish its confidence level via second-order correction (Arlot et al. 2010; Hao et al. 2019). The second order correction relies on the tail property of the reward distribution, in particular, the concentration behavior of $E(\boldsymbol{x}_a, \mathcal{F}_{t-1})$. We define a function $\beta(\cdot)$ to characterize the concentration behavior.

Definition 3. *Define $\beta(\cdot)$ as a function such that*

$$\mathbb{P}[E(\boldsymbol{x}_a, \mathcal{F}_{t-1}) \geq \beta(\alpha)\|\boldsymbol{x}_a\|_{\boldsymbol{V}_{t-1}^{-1}}] \leq \alpha,$$

where $\mathcal{F}_{t-1}$ is deterministic.

For example, as derived in Lemma 1, when the reward follows a sub-Gaussian distribution, the function $\beta(\alpha)$ has the expression $\beta(\alpha) = O(\ln(1/\alpha))$. For other distributions with tail heavier than the sub-Gaussian distribution (Bubeck, Cesa-Bianchi, and Lugosi 2013), we may have $\beta(\alpha) = O(\texttt{poly}(1/\alpha))$. Furthermore, $\beta(\alpha)$ can be made to infinity, i.e., $\beta(\alpha) = \exp(1/\alpha)$ or $\beta(\alpha) = \infty$, to characterize reward distribution with heavier tails. Namely, $\beta(\alpha)$ can characterize the full design space of reward distribution.

Theorem 4. *Suppose Condition 1 holds and $\mathcal{F}_t$ is deterministic. Suppose the $W_{a,t}, \forall a, t$, follow symmetric distribution. Then we have:*

$$\mathbb{P}\left[E(\boldsymbol{x}_a, \mathcal{F}_{t-1}) \geq \widehat{q}_t(\boldsymbol{x}_a, \widehat{\boldsymbol{\epsilon}}, 1 - \alpha(1-\delta)) + \Delta(t, \alpha)\right] \leq 2\alpha,$$

where $\delta \in (0, 1)$,

$$\Delta(t, \alpha) \triangleq \beta\left(\frac{\alpha}{|\mathcal{A}|}\right)\sqrt{2\sum_{s=1}^{t}(\boldsymbol{x}_a^T \boldsymbol{V}_t^{-1} \boldsymbol{x}_{a_s})^2 \|\boldsymbol{x}_{a_s}\|^2_{\boldsymbol{V}_t^{-1}} \ln\frac{1}{\alpha\delta}},$$

and $\boldsymbol{x}_a$ is dependent of $\boldsymbol{x}_{a_1}, \ldots, \boldsymbol{x}_{a_{t-1}}$. Furthermore, $\lim_{t \to \infty} \Delta(t, \alpha) = 0$ holds for any fixed α.

Remark: Theorem 4 states the sufficient conditions and a second order correction term $\Delta(t, \alpha)$, under which the quantile estimator $\widehat{q}_t(\boldsymbol{x}_a, \widehat{\boldsymbol{\epsilon}}, \alpha)$ can provide confidence level. Furthermore, for each fixed confidence level, the second order correction term $\Delta(t, \alpha)$ vanishes. Comparing with Theorem 3, one can observe that Theorem 4 requires an extra condition that the reward distribution is symmetric. Note that this extra condition is not due to the contextual bandit, but instead from the bootstrapping technique. To the best of our knowledge, handling non-symmetric distribution for non-asymptotic convergence is an open problem (Arlot et al. 2010; Chernozhukov et al. 2014; Yang, Shang, and Cheng 2017).

Quantile Bootstrapping Algorithm Design

Even though the corrected quantile estimator derived in Theorem 4, i.e., $\widehat{q}_t(\boldsymbol{x}, \widehat{\boldsymbol{\epsilon}}, \alpha(1-\delta)) + \Delta(t, \alpha)$, has the desired property as stated in Definition 1, one should not directly implement the bootstrapping oracle `BootQuantile` as $\widehat{q}_t(\boldsymbol{x}, \widehat{\boldsymbol{\epsilon}}, \alpha(1-\delta)) + \Delta(t, \alpha)$. This is because this implementation may not guarantee Condition 1 to hold, which in turn may lead to Theorem 4 not to hold.

We will first refine $\widehat{q}_t(\boldsymbol{x}, \widehat{\boldsymbol{\epsilon}}, \alpha(1-\delta)) + \Delta(t, \alpha)$, and then use this refinement to design the bootstrapping oracle `BootQuantile` such that Condition 1 can be guaranteed. To facilitate the analysis, we define the following notation to quantify the norm of a set of arms:

$$\|\mathcal{A}_t\|_{\boldsymbol{V}_{t-1}^{-1}} \triangleq \max_{a \in \mathcal{A}_t} \|\boldsymbol{x}_a\|_{\boldsymbol{V}_{t-1}^{-1}}.$$

Based on this notation, the following lemma states a sufficient condition to guarantee Condition 1.

Lemma 2. *Suppose it holds that*

$$\dim(\mathcal{A}_t) = \dim(\mathcal{A}), \qquad \forall t, \tag{3}$$

Then $\limsup_{t\to\infty} \|\mathcal{A}_t\|_{\boldsymbol{V}_{t-1}^{-1}} = 0$ *implies Condition 1.*

To show lemma 2, the following lemma derives a sufficient condition for Condition 1.

Lemma 3. *Suppose Eq. (3) holds. We have:*

$$\limsup_{t\to\infty} \|\mathcal{A}_t\|_{\boldsymbol{V}_{t-1}^{-1}} > 0 \Rightarrow \liminf_{t\to\infty} \|\mathcal{A}_t\|_{\boldsymbol{V}_{t-1}^{-1}} > 0.$$

Lemma 3 states that if $\liminf_{t\to\infty} \|\mathcal{A}_t\|_{\boldsymbol{V}_{t-1}^{-1}} > 0$ does not hold, then Condition 1 holds. It is easier to show that $\liminf_{t\to\infty} \|\mathcal{A}_t\|_{\boldsymbol{V}_{t-1}^{-1}} > 0$ leads to contraction than directly showing $\limsup_{t\to\infty} \|\mathcal{A}_t\|_{\boldsymbol{V}_{t-1}^{-1}} = 0$. The following theorem states the sufficient conditions under which $\liminf_{t\to\infty} \|\mathcal{A}_t\|_{\boldsymbol{V}_{t-1}^{-1}} > 0$ leads to contradiction, and it further refines the quantile estimator.

Theorem 5. *Suppose Eq (3) holds. Suppose α_t goes to zero as t goes to infinity. If `BootQuantile`$(\boldsymbol{x}, \mathcal{H}_{t-1}, 1-\alpha_t)$ returns $\widetilde{q}_t(\boldsymbol{x}, \widehat{\boldsymbol{\epsilon}}, 1 - \frac{\alpha_t}{2}(1-\delta))$, Algo. 1 guarantees Condition 1, where $\widetilde{q}_t(\boldsymbol{x}, \widehat{\boldsymbol{\epsilon}}, \alpha)$ is defined as an refined quantile estimator*

$$\widetilde{q}_t(\boldsymbol{x}, \widehat{\boldsymbol{\epsilon}}, \alpha) \triangleq \Delta\left(t, \frac{1-\alpha}{1-\delta}\right) + \begin{cases} +\infty, & \textit{if } \dim(\{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_{a_t}\}) > \dim(\{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_{a_{t-1}}\}), \\ \min\left\{\boldsymbol{x}^T \widehat{\boldsymbol{\theta}}_t + \widehat{q}_t(\boldsymbol{x}, \widehat{\boldsymbol{\epsilon}}, \alpha), 2SL\right\} - \boldsymbol{x}^T \widehat{\boldsymbol{\theta}}_t, & \textit{otherwise}, \end{cases}$$

where $\|\boldsymbol{x}_a\| \le S, \forall a \in \mathcal{A}$.

Remark: Theorem 5 states sufficient conditions on the action set $\mathcal{A}_t$ the confidence parameter α_t and a refined quantile estimator, such that Condition 1 holds. The condition on α_t means that α_t vanishes, which is commonly hold. The condition on action set $\mathcal{A}_t$ is that the dimension of the linear space spanned by $\mathcal{A}_t$ is the same as that spanned by $\mathcal{A}$. The main purpose of this condition is to make the proof and the statement of Theorem 5 clean. The refined estimator $\widetilde{q}_t(\boldsymbol{x}, \widehat{\boldsymbol{\epsilon}}, \alpha)$ means that we should always give the highest priority to those arms whose feature vector can increase the dimension of the linear space spanned by the feature vectors of historical actions. The following lemma states that we can still provide confidence level for the refined quantile estimator.

Lemma 4. *The refined quantile estimator $\widetilde{q}_t(\boldsymbol{x}, \widehat{\boldsymbol{\epsilon}}, \alpha)$ has the following confidence level:*

$$\mathbb{P}\left[E(\boldsymbol{x}_a, \mathcal{F}_{t-1}) \le \widetilde{q}_t\left(\boldsymbol{x}, \widehat{\boldsymbol{\epsilon}}, 1 - \frac{\alpha}{2}(1-\delta)\right)\right] \ge 1 - \alpha,$$

where $\mathcal{F}_t$ is deterministic.

Lemma 4 provides confidence level for the refined quantile estimator. Namely, the refined quantile estimator has the nice properties defined in Definition 1. Thus, we use this refined quantile estimator to design our bootstrapping oracle. Algorithm 2 outlines the procedures to compute the refined oracle. The key step of Algorithm 2 is step 4, which computes the estimator $\widehat{q}_t(\boldsymbol{x}_a, \widehat{\boldsymbol{\epsilon}}, \alpha_t(1-\delta))$. It is computationally expensive to compute the exact value for $\widehat{q}_t(\boldsymbol{x}_a, \widehat{\boldsymbol{\epsilon}}, \alpha_t(1-\delta))$. In the implementation, one can use Monte Carlo simulation to approximate it, which is quite common in multiplier bootstrapping literature (Arlot et al. 2010; Hao et al. 2019). The following theorem derive the regret as:

Theorem 6. *Suppose Eq. (3) holds. Under the condition of Theorem 1 and Algo. 2, we have*

$$\begin{aligned}&\mathcal{R}_T(\mathbb{A}_{\texttt{BootLinUCB}}) \\ &\le \sum_{t=1}^{\dim(\mathcal{A})} g_t + 4 \sum_{t=\dim(\mathcal{A})+1}^{T} \binom{t+A-2}{A-1} A\alpha_t g_t \\ &\quad + \sum_{t=\dim(\mathcal{A})+1}^{T} \mathbb{E}\left[\widetilde{q}_t(\boldsymbol{x}_{a_t}, \widehat{\boldsymbol{\epsilon}}, 1-\alpha_t) - \widetilde{q}_t(-\boldsymbol{x}_{a_t}, \widehat{\boldsymbol{\epsilon}}, 1-\alpha_t)\right].\end{aligned}$$

Furthermore, suppose the reward follows sub-Gaussian distribution, and $\alpha_t = 1/(\binom{t+A-2}{A-1}t)$, then the above regret has an order of $\mathcal{R}_T(\mathbb{A}_{\texttt{BootLinUCB}}) \le O(A\sqrt{T}(\ln T)^2)$.

Remark: Theorem 6 states a general regret upper bound for our bootLinUCB algorithm under the refined quantile

Algorithm 2 `BootQuantile`$(\boldsymbol{x}_a, \mathcal{H}_{t-1}, 1-2\alpha_t)$

```
1: $a \leftarrow \boldsymbol{x}_a^T \widehat{\boldsymbol{\theta}}_t$.
2: Compute $\widehat{\boldsymbol{\epsilon}}$ from the reward history $\mathcal{H}_{t-1}$
3: $c \leftarrow \Delta(t, \alpha_t)$ .
4: $b \leftarrow \widehat{q}_t(\boldsymbol{x}_a, \widehat{\boldsymbol{\epsilon}}, \alpha_t(1-\delta))$
5: if $\dim(\{\boldsymbol{x}_{a_1}, \ldots, \boldsymbol{x}_{a_{t-1}}\}) < \dim(\{\boldsymbol{x}_{a_1}, \ldots, \boldsymbol{x}_{a_t}\})$ then
6:     RETURN $+\infty$.
7: else
8:     RETURN $\min\{a+b, 2SL\} - a + c$
9: end if
```

estimator. This regret upper bound can be further simplified to be sub-linear if the reward follows sub-Gaussian distribution.

Experiments on Synthetic Data

Experiment setting. We compare our BootLinUCB algorithm with the latest bootstrapping based LinUCB algorithm (Hao et al. 2019) and the classical LinUCB algorithm (Chu et al. 2011). Since the latest bootstrapping based LinUCB algorithm (Hao et al. 2019) does not have theoretical guarantee, we denote it by BootLinHeu, where Heu represents heuristic.

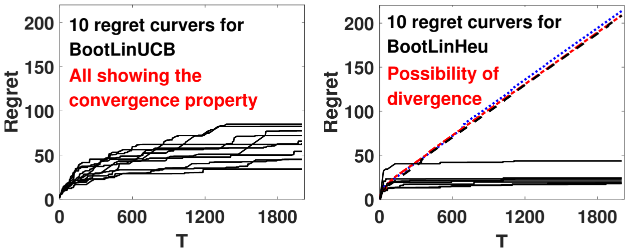
Consider $T = 2000$ decision rounds. The $W_{a,t}, \forall a, t$ follow normal distribution with mean 0 and variance σ. We generate the feature vectors of A arms as follows: (1) generate $\min\{d, A\}$ orthogonal feature vectors with unit square norm (details refer to our code); (2) each of the remaining $A - \min\{d, A\}$ feature vectors is drawn from $[0, 1]^d$ uniformly at random. The preference parameter $\boldsymbol{\theta}_*$ is drawn from $[0, 1]^d$ uniformly at random. In each round, we set $\mathcal{A}_t = \mathcal{A}$. Similar with previous works (Arlot et al. 2010; Hao et al. 2019), we use Monte Carlo simulation to estimate the quantile estimator $\widehat{q}_t(\boldsymbol{x}, \widehat{\boldsymbol{\epsilon}}, \alpha)$ with 1000 simulation rounds. We set $\alpha_t = 1/\sqrt{t+2}, \delta = 1/(t+2)$ for our BootLinUCB, and set $\alpha_t = 1/\sqrt{t+2}$ for the LinUCB algorithm. We set parameters for the BootLinHeu algorithm according to (Hao et al. 2019). Unless we state explicitly, we consider the following default parameters: $A = 20$ arms, features with $d = 10$ dimension, regularization parameter $\lambda = 1$, reward variance $\sigma = 1$. For each algorithm, we repeatedly run it for 100 times, and calculate its average regret over 100 times.

Convergence comparison. We first use a specific setting to show that the BootLinHeu (Hao et al. 2019) has a risk of diverging, while the BootLinUCB algorithm does not have this risk. We set $A = 10$ and $d = 5$. Five of the feature vectors are five standard base vectors, and the remaining five as well as $\boldsymbol{\theta}_*$ are:

$(0.08, 0.32, 0.22, 0.14, 0.73)$, $(0.1, 0.22, 0.15, 0.09, 0.68)$,
$(0.58, 0.87, 0.32, 0.3, 0.14)$, $(0.18, 0.88, 0.83, 0.24, 0.65)$,
$(0.86, 0.2, 0.51, 0.83, 0.97)$,
$\boldsymbol{\theta}_*$=$(0.23, 0.36, 0.61, 0.26, 0.73)$.

Figure 1 shows ten regret curves for the BootLinUCB and BootLinHeu algorithm respectively. One can observe that all regret curves of BootLinUCB are flat, while three out of ten regret curves of BootLinHeu increases linearly in t. This implies that the BootLinHeu has a risk of diverging (i.e., always select an sub-optimal arm leading to regret increases linearly in t), while our BootLinUCB algorithm does not have this risk. Thus, in remaining experiments we do not further compare with BootLinHeu.

Mismatch of reward tail distribution. Now we study the robustness of our BootLinUCB algorithm with respect to the mismatching of reward distribution tail. We input the variance of reward distribution $\sigma_{in} = 1$ to both LinUCB and BootLinUCB. Figure 2 shows the average regret of LinUCB and BootLinUCB as we vary the ground truth variance σ from 0.1 to 0.5. One can observe that when $\sigma = 0.1$, the average regret of BootLinUCB is around half of LinUCB (T=2000). Increasing the ground truth variance to $\sigma = 0.5$, the average regret of BootLinUCB is around $100/160 = 5/8$ of LinUCB (T=2000). These results further confirm that the BootLinUCB can significantly reduce the over exploration problem LinUCB caused by mismatching of the reward tail distribution.



(a) BootLinUCB (b) BootLinHeu

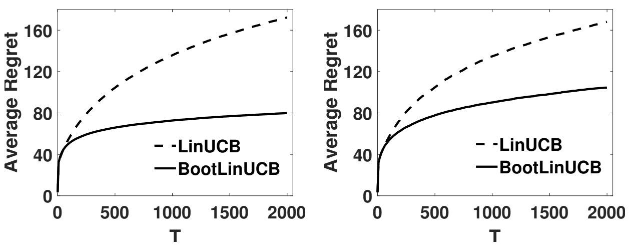
Figure 1: Regret curves of BootLinUCB and BootLinHeu.

Conclusion

This paper presents the BootLinUCB algorithm for the contextual bandit. BootLinUCB is a data driven UCB based algorithm, which uses the multiplier bootstrapping technique to estimate the UCB of contextual from the historical rewards. The BootLinUCB is more robust to misspecification of the reward tail distribution than the previous reward tail distribution based UCB algorithms in contextual bandit. In particular, we design an estimator for the UCB of contextual bandit with theoretical guarantee on the convergence. Based on the estimator we design our BootLinUCB algorithm. We also prove that the BootLinUCB has a sub-linear regret upper bound and conduct extensive experiments to show its superior performance over a variety of baselines. Our approach open doors for others to consider similar bootstrapping technique for other online learning algorithms or reinforcement algorithms.

Acknowledgments

This work was supported in part by Chongqing Natural Science Foundation (cstc2020jcyj-msxmX0652), Chongqing High-Technology Innovation and Application Development Funds (cstc2019jscx-msxmX0422, cstc2019jscx-fxyd0385) and the Fundamental Research Funds for the Central Universities (2020CDJ-LHZZ-057). Hong Xie is the corresponding author.



(a) $\sigma = 0.1$ (b) $\sigma = 0.5$

Figure 2: Regret under mismatch of reward tail distribution.

References

Abbasi-Yadkori, Y.; Antos, A.; and Szepesvári, C. 2009. Forced-exploration based algorithms for playing in stochastic linear bandits. In *COLT Workshop on On-line Learning with Limited Feedback*, volume 91, 235.

Abbasi-Yadkori, Y.; Pál, D.; and Szepesvári, C. 2011. Improved algorithms for linear stochastic bandits. In *Advances in Neural Information Processing Systems*, 2312–2320.

Agrawal, S.; and Goyal, N. 2013. Thompson sampling for contextual bandits with linear payoffs. In *International Conference on Machine Learning*, 127–135.

Arlot, S.; Blanchard, G.; Roquain, E.; et al. 2010. Some nonasymptotic results on resampling in high dimension, I: confidence regions. In *The Annals of Statistics*, volume 38, 51–82. Institute of Mathematical Statistics.

Auer, P. 2002. Using confidence bounds for exploitation-exploration trade-offs. In *Journal of Machine Learning Research*, volume 3, 397–422.

Bubeck, S.; Cesa-Bianchi, N.; and Lugosi, G. 2013. Bandits with heavy tail. In *IEEE Transactions on Information Theory*, volume 59, 7711–7717. IEEE.

Chernozhukov, V.; Chetverikov, D.; Kato, K.; et al. 2014. Gaussian approximation of suprema of empirical processes. In *The Annals of Statistics*, volume 42, 1564–1597. Institute of Mathematical Statistics.

Chu, W.; Li, L.; Reyzin, L.; and Schapire, R. 2011. Contextual bandits with linear payoff functions. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, 208–214.

Eckles, D.; and Kaptein, M. 2014. Thompson sampling with the online bootstrap. In *arXiv preprint arXiv:1410.4009*.

Elmachtoub, A. N.; McNellis, R.; Oh, S.; and Petrik, M. 2017. A practical method for solving contextual bandit problems using decision trees. In *arXiv preprint arXiv:1706.04687*.

Filippi, S.; Cappe, O.; Garivier, A.; and Szepesvári, C. 2010. Parametric bandits: The generalized linear case. In *Advances in Neural Information Processing Systems*, 586–594.

Gampa, P.; and Fujita, S. 2019. BanditRank: Learning to Rank Using Contextual Bandits. In *arXiv preprint arXiv:1910.10410*.

Glowacka, D.; et al. 2019. Bandit algorithms in information retrieval. In *Foundations and Trends® in Information Retrieval*, volume 13, 299–424. Now Publishers, Inc.

Hao, B.; Yadkori, Y. A.; Wen, Z.; and Cheng, G. 2019. Bootstrapping Upper Confidence Bound. In *Advances in Neural Information Processing Systems*, 12123–12133.

Hofmann, K.; Whiteson, S.; de Rijke, M.; et al. 2011. Contextual bandits for information retrieval. In *NIPS 2011 Workshop on Bayesian Optimization, Experimental Design, and Bandits, Granada*, volume 12, 2011.

Krishnamurthy, A.; Wu, Z. S.; and Syrgkanis, V. 2018. Semiparametric contextual bandits. In *arXiv preprint arXiv:1803.04204*.

Kveton, B.; Szepesvari, C.; Vaswani, S.; Wen, Z.; Lattimore, T.; and Ghavamzadeh, M. 2019. Garbage In, Reward Out: Bootstrapping Exploration in Multi-Armed Bandits. In Chaudhuri, K.; and Salakhutdinov, R., eds., *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, 3601–3610. Long Beach, California, USA: PMLR.

Lattimore, T.; and Szepesvári, C. 2020. *Bandit algorithms*. Cambridge University Press.

Li, L.; Chu, W.; Langford, J.; and Schapire, R. E. 2010. A contextual-bandit approach to personalized news article recommendation. In *Proceedings of the 19th international conference on World wide web*, 661–670. ACM.

Osband, I.; and Van Roy, B. 2015. Bootstrapped thompson sampling and deep exploration. In *arXiv preprint arXiv:1507.00300*.

Ouyang, T.; Li, R.; Chen, X.; Zhou, Z.; and Tang, X. 2019. Adaptive User-managed Service Placement for Mobile Edge Computing: An Online Learning Approach. In *IEEE INFOCOM 2019-IEEE Conference on Computer Communications*, 1468–1476. IEEE.

Sudarsanam, N.; and Ravindran, B. 2016. Linear Bandit algorithms using the Bootstrap. In *arXiv preprint arXiv:1605.01185*.

Tang, L.; Jiang, Y.; Li, L.; Zeng, C.; and Li, T. 2015. Personalized recommendation via parameter-free contextual bandits. In *Proceedings of the 38th international ACM SIGIR conference on research and development in information retrieval*, 323–332.

Vaswani, S.; Kveton, B.; Wen, Z.; Rao, A.; Schmidt, M.; and Abbasi-Yadkori, Y. 2018. New insights into bootstrapping for bandits. In *arXiv preprint arXiv:1805.09793*.

Wang, H.; Wu, Q.; and Wang, H. 2016. Learning Hidden Features for Contextual Bandits. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*, CIKM '16, 1633–1642. New York, NY, USA: ACM. ISBN 978-1-4503-4073-1. doi: 10.1145/2983323.2983847. URL http://doi.acm.org/10.1145/2983323.2983847.

Yang, Y.; Shang, Z.; and Cheng, G. 2017. Non-asymptotic theory for nonparametric testing. In *arXiv preprint arXiv:1702.01330*.

Zhang, X.; Xie, H.; Li, H.; and Lui, J. 2019. Toward Building Conversational Recommender Systems: A Contextual Bandit Approach. In *arXiv preprint arXiv:1906.01219*.

Zhu, F.; Zhu, X.; Wang, S.; Yao, J.; and Huang, J. 2017. Robust Contextual Bandit via the Capped Ell Two Norm. In *arXiv preprint arXiv:1708.05446*.