Improved Algorithms for Conservative Exploration in Bandits

Evrard Garcelon,[1] Mohammad Ghavamzadeh,[1] Alessandro Lazaric,[1] Matteo Pirotta[1]
[1]Facebook AI Research,
evrard.garcelon@gmail.com, {mgh, lazaric, pirotta}@fb.com

Abstract

In many fields such as digital marketing, healthcare, finance, and robotics, it is common to have a well-tested and reliable baseline policy running in production (e.g., a recommender system). Nonetheless, the baseline policy is often suboptimal. In this case, it is desirable to deploy online learning algorithms (e.g., a multi-armed bandit algorithm) that interact with the system to learn a better/optimal policy *under the constraint* that during the learning process the performance is almost never worse than the performance of the baseline itself. In this paper, we study the *conservative learning* problem in the contextual linear bandit setting and introduce a novel algorithm, the Conservative Constrained LinUCB (CLUCB2). We derive regret bounds for CLUCB2 that match existing results and empirically show that it outperforms state-of-the-art conservative bandit algorithms in a number of synthetic and real-world problems. Finally, we consider a more realistic constraint where the performance is verified only at predefined checkpoints (instead of at every step) and show how this relaxed constraint favorably impacts the regret and empirical performance of CLUCB2.

Introduction

Many problems in fields such as digital marketing, healthcare, finance, and robotics can be formulated as decision-making under uncertainty. Although many learning algorithms have been developed to find a good/optimal policy for these problems, a major obstacle in using them in real-world applications is the lack of guarantees for the actual performance of the policies they execute over time. Therefore, for the applicability of these algorithms, it is important that they execute policies that are guaranteed to perform at least as well as an existing *baseline*. We can think of the baseline either as a baseline value or the performance of a baseline policy. It is important to note that since the learning algorithms generate these polices from data, they are random variables, and thus, all the guarantees on their performance should be in high probability. This problem has been recently studied under the general title of *safety w.r.t. a baseline* in bandits and reinforcement learning (RL), in both *offline* (Bottou et al. 2013; Thomas, Theocharous, and Ghavamzadeh 2015a; 2015b; Swaminathan and Joachims 2015; Petrik, Ghavamzadeh, and Chow 2016) and *online* (Mansour, Slivkins, and Syrgkanis 2015; Wu et al. 2016; Kazerouni et al. 2017; Katariya et al. 2019) settings.

In the online setting, which is the focus of this paper, the learning algorithm updates its policy while interacting with the system. Although the algorithm eventually learns a good or an optimal policy, there is no guarantee on the performance of the intermediate policies, especially at the very beginning, when the algorithm needs to heavily explore different options. Therefore, in order to make sure that at any point in time the (cumulative) performance of the policies generated by the algorithm is not worse than the baseline, it is important to control the exploration and make it more *conservative*. Consider a recommender system that runs our learning algorithm. Although we are confident that our algorithm will eventually learn a strategy that performs as well as the baseline, and possibly even better, we should control its exploration not to lose too many customers, as a result of providing them with unsatisfactory recommendations. This setting has been studied in multi-armed bandits (Wu et al. 2016), contextual linear bandits (Kazerouni et al. 2017), and stochastic combinatorial semi-bandits (Katariya et al. 2019). These papers formulate the problem using a constraint defined based on the performance of the baseline policy (mean of the baseline arm in the multi-armed bandit case), and modify the corresponding UCB-type algorithm (Auer, Cesa-Bianchi, and Fischer 2002a) to satisfy this constraint. At each round, the conservative bandit algorithm computes the action suggested by the corresponding UCB algorithm, if the action satisfies the constraint, it is taken, otherwise, the algorithm acts according to the baseline policy. Another algorithm in the online setting is by (Mansour, Slivkins, and Syrgkanis 2015) that balances exploration and exploitation such that the actions taken are compatible with the agent's (customer's) incentive formulated as a Bayesian prior.

In this paper, we focus on UCB-type algorithms and improve the design and empirical performance of the conservative algorithms in the contextual linear bandit setting. We first highlight the limitations of the existing conservative bandit algorithms (Wu et al. 2016; Kazerouni et al. 2017) and show that simple modifications in constructing the conservative condition and the arm-selection strategy may significantly improve their performance. We show that our algorithm is formally correct by proving regret bound, matching existing results and illustrate its practical advantage w.r.t. state-of-the-art algorithms in a number of synthetic and real-world environments. Finally, we consider the more realistic scenario where the conservative constraint is verified at predefined checkpoints (e.g., a manager may be interested in verifying the performance of the learning algorithm every few days). In this case, we prove a regret bound showing that as the checkpoints become less frequent, the conservative condition has less impact on the regret, which eventually reduces to the standard (unconstrained) one.

Conservative Contextual Linear Bandits

We consider the standard linear bandit setting. At each time t, the agent selects an arm $a_t \in \mathcal{A}_t$ and observes a reward

$$r_a^t = \langle \theta^\star, \phi_a^t \rangle + \eta_a^t := \mu_a^t + \eta_a^t, \qquad (1)$$

where $\theta^\star \in \mathbb{R}^d$ is a parameter vector, $\phi_a^t \in \mathbb{R}^d$ are the features of arm a at time t, and η_a^t is a zero-mean σ^2-subgaussian noise. When the features correspond to the canonical basis, this formulation reduces to multi-armed bandit (MAB) with d arms. In the more general case, the features may depend on a context x_t, so that $\phi_a^t = \phi(x_t, a)$ denotes the feature vector of a context-action pair (x_t, a) and (1) defines the so-called linear contextual bandit setting.

We rely on the following standard assumption on the features and the unknown parameter $\theta^\star$.

Assumption 1. *There exist $B, D \geq 0$, such that $\|\theta^\star\|_2 \leq B$, $\|\phi_a^t\| \leq D$, and $\langle \theta^\star, \phi_a^t \rangle \in [0, 1]$, for all t and a.*

Given a finite horizon n, the performance of the agent is measured by its (pseudo)-*regret*:

$$R(n) = \sum_{t=1}^{n} \langle \theta^\star, \phi_{a^\star}^t \rangle - \langle \theta^\star, \phi_{a_t}^t \rangle,$$

where $a_t^\star \in \arg\max_a \langle \theta^\star, \phi_a^t \rangle$ is the optimal action at time t.

In the conservative setting, the objective is to minimize the regret under additional performance constraints w.r.t. a known baseline. We assume the agent has access to a *baseline policy*, which selects action b_t at time t.[1] The learning problem is constrained such that, at any time t, the difference in performance (i.e., expected cumulative reward) between the baseline and the agent should never fall below a predefined fraction of the baseline performance. Formally, the *conservative constraint* is given by

$$\forall t > 0, \qquad \sum_{i=1}^{t} \mu_{a_i}^i \geq (1 - \alpha) \sum_{i=1}^{t} \mu_{b_i}^i, \qquad (2)$$

where $\alpha \in (0, 1)$ is the conservative level. As the LHS of (2) is a random variable depending on the agent's strategy, we require this constraint to be satisfied with high probability. Finally, in order to keep the presentation and analysis simple, we rely on the following assumption.

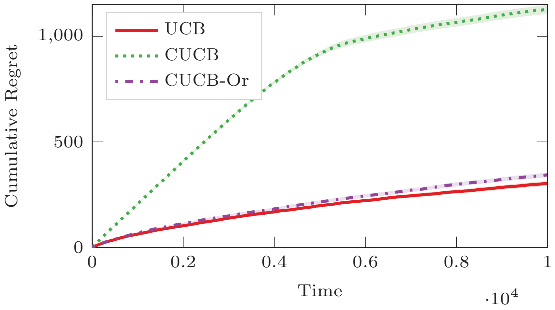

Figure 1: Comparison of the cumulative regret between UCB, its conservative variant (CUCB), and an *oracle* version of CUCB, where (2) can be evaluated exactly to decide whether to select the UCB arm or the baseline.

Assumption 2. *For any $t > 0$, the performance of the baseline strategy until t is known, i.e., $\sum_{i=1}^{t} \mu_{b_i}^i$ can be evaluated by the agent.*

This assumption is often reasonable since the baseline performance can be estimated from historical data (see Rem. 3 in (Kazerouni et al. 2017)). Furthermore, as shown in (Wu et al. 2016; Kazerouni et al. 2017), this knowledge can be removed and the algorithm can be modified to incorporate the estimation process (and preserve the same order of regret).

Conservative Exploration. Conservative exploration algorithms (Wu et al. 2016; Kazerouni et al. 2017) are based on a two-step process to select the action to play. In the first step, they compute an optimistic action based on the optimism-in-the-face-of-uncertainty principle, i.e., using UCB (Auer, Cesa-Bianchi, and Fischer 2002b) or LinUCB (Li et al. 2010; Abbasi-Yadkori, Pál, and Szepesvári 2011), which is effective in exploring and minimizing the regret over time. In the second step, they evaluate the conservative condition by replacing the unknown mean with a statistical lower bound. If the condition is verified, they play the optimistic action, otherwise, they act conservatively by selecting the baseline b_t. Playing the baseline over multiple steps contributes to "build a conservative budget", so that condition (2) is more likely to be satisfied by the UCB arm, and thus, allowing to execute explorative actions.

Formally, let S_t^b be the set of times up to t (included) where the agent played the baseline and $S_{t-1} = [t] \setminus S_t^b$ be the complementary set, i.e., when the agent played the UCB action. CLUCB uses the information collected when playing non-conservatively to build an estimator of $\theta^\star$ by solving a regularized least-square problem $\widehat{\theta}_t = (\Phi_t \Phi_t^\top + \lambda I)^{-1} \Phi_t Y_t$, where $\lambda > 0$, $\Phi_t = (\phi_{a_i}^i)_{i \in S_{t-1}} \in \mathbb{R}^{d \times |S_{t-1}|}$ and $Y_t = (r_{a_i}^i)_{i \in S_{t-1}} \in \mathbb{R}^{|S_{t-1}|}$. Denote by $V_t = \lambda I + \Phi_t \Phi_t^\top$ the design matrix of the regularized least-square problem and by $\|x\|_V = \sqrt{x^\top V x}$ the weighted norm w.r.t. any positive matrix $V \in \mathbb{R}^{d \times d}$. We define the confidence set

[1] In the non-contextual case, the baseline policy reduces to a single baseline action b.

$\Theta_t = \{\theta \in \mathbb{R}^d : \|\theta - \widehat{\theta}\|_{V_t^{-1}} \le \beta_t\}$ where

$$\beta_t = \sigma\sqrt{d\log\left(\frac{1+D^2(1+|S_{t-1}|)/\lambda}{\delta}\right)} + B\sqrt{\lambda}, \qquad (3)$$

which guarantees that $\theta^\star \in \Theta_t$, for all $t > 0$, w.p. $1-\delta$.

Similar to LinUCB, the optimistic action is computed as

$$a_t \in \arg\max_{a\in\mathcal{A}_t} \max_{\theta\in\Theta_t} \langle \theta, \phi^t_a \rangle,$$

and CLUCB decides if the action is "safe" by evaluating the following conservative condition:

$$\sum_{i\in S^b_{t-1}} \mu^i_{b_i} + \min_{\theta\in\Theta_t} \left\langle \theta, \phi^t_{a_t} + \sum_{i\in S_{t-1}} \phi^i_{a_i} \right\rangle \ge (1-\alpha)\sum_{i=1}^{t} \mu^i_{b_i}. \qquad (4)$$

The leftmost term in (4) represents the expected cumulative reward associated to baseline actions played up to time $t-1$. The second term –minimization problem– denotes the lower bound to the cumulative reward of optimistic actions. Condition 4 amounts to evaluate a lower bound on the cumulative reward thanks to the confidence interval constructed up until the current time. If the condition is satisfied, then the optimistic action a_t is played, otherwise, the baseline strategy is selected and the corresponding action b_t is executed.

Limitations. While CLUCB enjoys strong regret guarantees (in the MAB setting, it is indeed near-optimal), its empirical behavior is often over-conservative, i.e., the baseline strategy is selected for a very long time to build enough *conservative budget* before the actual exploration takes place. We identify two main algorithmic causes for such behavior. First, in building the conservative condition (4), CUCB and CLUCB rely on possibly loose statistical lower-bounds for the mean of the actions selected so far. This is well illustrated by the simulation in Fig. 1 in the MAB setting, where we report the performance of UCB, CUCB, and an *oracle* variant of CUCB, when the conservative condition (2) is evaluated exactly (i.e., no lower-bound is used). While the *oracle* version has almost the same regret as UCB, and thus, showing that the conservative condition itself does not have a major impact on the exploration of UCB, CUCB has a much higher regret. This shows that possibly loose estimates of the conservative condition have a significant impact on the regret. In fact, tightening the conservative condition would allow selecting the baseline strategy only when it is "strictly" needed, thus, reducing the conservative steps and improve the overall exploration performance.

Second, the two-step selection strategy of CUCB and CLUCB performs either an exploration step, when the UCB action is selected, or a conservative step, when the baseline is executed. Such sharp division between exploration and conservative steps may be unnecessary, as other actions may still be "safe" (i.e., satisfying the conservative condition), and thus, contribute to build "conservative budget", and, at the same time, be useful for exploration (e.g., optimistic), despite not being the UCB action. Exploiting such arms may lead to a better performance. Finally, the conservative condition (2) itself is often too strict in practice. Instead of performing almost as well as the baseline *at every step*, it is more likely that recurrent "checkpoints" are set at which the agent is required to meet the condition. In this case, the agent may have extra time to perform exploratory actions and possibly recover from bad past choices when getting close to the conservative checkpoint.

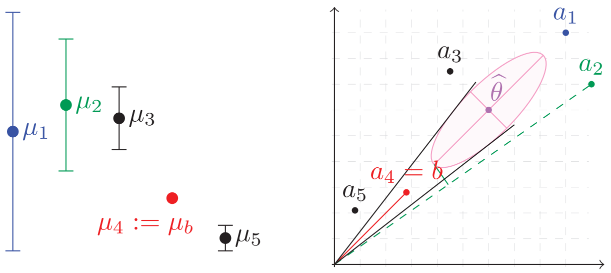

Figure 2: Examples of settings where the UCB arm (blue) does not satisfy the conservative condition but there is another "safe" arm to play (rather than the baseline).

In the next section, we address the two algorithmic limitations described above, while we later illustrate how a relaxed conservative condition may indeed allow an agent to achieve much smaller regret.

Improved Conservative Exploration

In this section we present Conservative Constrained LinUCB (CLUCB2) (Alg. 1 with $T = 1$), an improved conservative exploration algorithm for contextual linear bandit. All the proofs can be found in the extended version.

CLUCB2

The first improvement w.r.t. CLUCB is relative to the conservative condition (4). When evaluating the rewards accumulated by the agents so far, we rely on the fact that the sequence $(r^i_{a_i} - \mu^i_{a_i})_{i\in S_{t-1}}$ is a Martingale Difference Sequence (MDS) with respect to the filtration $\mathcal{F}_{t-1} = \sigma\left((a_j, \phi^j_{a_j}, r^j_{a_j})_{j\in S_{t-1}}\right)$, i.e., the history strictly before time t. Indeed, the choice of arm a_i is $\mathcal{F}_i$-measurable and for all $i \in S_{t-1}$:

$$\mathbb{E}\left[r^i_{a_i} - \mu^i_{a_i} \mid \mathcal{F}_i\right] = \sum_{j=1}^{|\mathcal{A}_i|} \mathbb{1}\left(a_i = j\right)\left(\mathbb{E}\left[r^i_j\right] - \mu^i_j\right) = 0$$

By using Freedman's inequality (Freedman 1975) for martingales, with probability at least $1-\delta$ we have

$$\Big|\sum_{i\in S_{t-1}} (r^i_{a_i} - \mu^i_{a_i})\Big| \le \psi_L(t) := \sigma\sqrt{2|S_{t-1}|L^\delta_t} + \frac{2}{3}L^\delta_t \qquad (5)$$

where $L^\delta_t := \log\left(3(|S_{t-1}| \vee 1)^2/\delta\right)$. Thus we replace (4) by

$$\sum_{i\in S^b_{t-1}} \mu^i_{b_i} + \sum_{i\in S_{t-1}} r^i_{a_i} - \psi_L(t) + \min_{\theta\in\Theta_t} \langle \theta, \phi^t_{a_t} \rangle \ge (1-\alpha)\sum_{i=1}^{t} \mu^i_{b_i}. \qquad (6)$$

While it is not possible to prove that (6) is always tighter, in the next section we provide an extensive discussion on the potential improvements.

A second limitation of CLUCB is its two-step approach to action selection, in which either the optimistic action satisfies the conservative condition or the baseline b_t is selected. The idea behind this strategy is that it is necessary to select baseline actions before exploring any other action in order to build a "conservative budget", which allows performing effective exploration later on (once the conservative condition is met). In CLUCB2 we propose to combine the explorative and conservative requirements by selecting the most optimistic (i.e., useful for exploration) "safe" (i.e., satisfying the conservative condition) action. Formally, the algorithm computes the set $\mathcal{C}_t$ of "safe" arms such that[2]

$$\mathcal{C}_t = \Big\{ a \in \mathcal{A}_t \setminus \{b_t\} \mid \sum_{i \in S_{t-1}} r^i_{a_i} - \psi_L(t) + \sum_{i \in S^b_{t-1}} \mu^i_{b_i} + \max\Big\{ \min_{\theta \in \Theta_t} \langle \theta, \phi^t_a \rangle, 0 \Big\} \geq (1-\alpha) \sum_{i=1}^{t} \mu^i_{b_i} \Big\} \quad (7)$$

where ψ_L is the Martingale bound given in Eq. 5. The algorithm plays the arm that solves the following constrained optimization problem:

$$a_t \in \arg\max \Big\{ r^t_{b_t}, \max_{a \in \mathcal{C}_t} \max_{\theta \in \Theta_t} \langle \theta, \phi^t_a \rangle \Big\} \quad (8)$$

where, by definition, the max over an empty set is $-\infty$. The maximizer is either the baseline arm b_t or an arm in $\mathcal{C}_t$ that is optimistic w.r.t. the baseline.

In order to illustrate the idea behind (8), consider the configuration illustrated in Fig. 2*(left)* for a MAB setting. If the algorithm has not built enough margin, the UCB arm (a_1) would not satisfy the conservative condition as its lower-confidence bound is well below the baseline. As a consequence, CUCB selects the baseline arm a_4. A direct improvement can be achieved by selecting arm a_3 (i.e., the one with the higher lower bound among the arms passing the conservative condition) as suggested in (Wu et al. 2016). Nonetheless, while arm a_3 is indeed better than baseline and it allows building conservative budget faster, it may not be effective from an exploration point of view. In (8) we suggest arm a_2 would be a better choice as it does not give up on reducing the regret (i.e., it has a larger UCB than a_2). This may indeed result in a better tradeoff between building conservative budget and performing effective exploration. Finally, Fig. 2*(right)* shows that (8) may be effective even in the linear setting. The stretched out ellipsoid on one axis gives a precise estimate of some bad arms, while good arms would not be selected by choosing the arm maximizing the lower bound. Even though the arms are more correlated due to the linear structure, the interpretation of this case is as the one for stochastic MABs.

From a computational perspective, the cost of an update for both CLUCB and CLUCB2 is $O(Ad^3)$. This complexity comes from the maximization over action and the construction of the confidence intervals. Compared to CLUCB, CLUCB2 has to evaluate the conservative condition for each arm instead of only for the UCB arm. However, the cost of this operation is dominated by the arm selection procedure.

Algorithm 1: CLUCB2 ($T = 1$) and CLUCB2T

```
Input: $\alpha$, $\delta$, $T$
Set $S_0 = S^b_0 = \emptyset$, $k = 0$
for $t = 1, \dots, n$ do
    Compute "safe" set $C_t$ as in Eq. 7 or Eq. 12
    Compute $a_t$ by solving (8)
    Pull arm $a_t$ and observe $r^t_{a_t}$
    if $a_t \neq b_t$ then
        Set $S_t = S_{t-1} \cup \{t\}$, $S^b_t = S^b_{t-1}$
        Compute new confidence set $\Theta_{t+1}$
    else
        Set $S_t = S_{t-1}$, $S^b_t = S^b_{t-1}$, $\Theta_{t+1} = \Theta_t$
    if $t \bmod T = 0$ then
        $k = k + 1$
```

Finally, we notice that following the same construction as in (Kazerouni et al. 2017), CLUCB2 can be easily adapted to the case when Asm. 2 does not hold and the baseline performance needs to be estimated online.

Theoretical Analysis

Let $\Delta^t_a = \mu^t_{a^\star_t} - \mu^t_a$ be the action gap at time t. As in (Kazerouni et al. 2017), we rely on the following assumption.

Assumption 3. *There exists* $0 \leq \Delta_l \leq \Delta_h$ *and* $0 < \mu_l \leq \mu_h$ *such that for every t :*

$$\Delta_l \leq \Delta^t_{b_t} \leq \Delta_h \quad \text{and} \quad \mu_l \leq \mu^t_{b_t} \leq \mu_h$$

Asm. 3 ensures that the baseline policy has a minimum level of performance, which is reasonable since the baseline policy is the strategy currently used by default. Note that in MABs and linear bandits $\mu_l = \mu_h = \mu_b$ since the performance does not depend on a system context. The terms Δ_l and μ_h are not critical quantities in the regret bound and it is possible to take $\Delta_l = 0$, $\mu_h = 1$. We are now ready to state the following result for CLUCB2.

Theorem 1. *For any contextual linear bandit problem, under Asm. 1, 2, and 3,* CLUCB2 *satisfies the conservative condition with probability* $1 - \delta$ *and its regret can be bounded for any* $n > 0$ *with probability at least* $1 - \delta$ *by*

$$R_{\text{CLUCB2}}(n) \leq O\Bigg(\sigma d \log\left(\frac{nD^2}{\lambda d}\right) \sqrt{n} + \frac{\Delta_h d^2}{(\alpha\mu_l)^2} \big(\sqrt{\lambda} B + \sigma\big)^2 \log\left(\frac{d^2\big(\sqrt{\lambda} B + \sigma\big)^2 \sqrt{D_0}}{\sqrt{\delta}\alpha\mu_l} \right)^2 \Bigg) \quad (9)$$

where $D_0 := \max\big\{2D^2/\lambda, 3\big\}$.

[2] S_{t-1} contains all steps when the baseline is *not* selected, and now it may include arms different from the UCB arm.

This regret is of the same order as the bound for CLUCB.[3] While this shows that the changes made to CLUCB are "safe", we cannot prove a direct improvement to the regret performance, apart from better constants (the regret of CLUCB is at least half of the one of CLUCB2, see appendix). Notice also that in the MAB case, this is not even possible in general, as CUCB is already proved to match the lower bound (in a worst-case sense). However, a worst-case argument may be misleading in the ranking of the algorithms.

The empirical validation reported in the experimental section will provide a more direct evidence of the improvement of CLUCB2 over CLUCB. In the rest of the section we analyze the parts of the regret that are most directly impacted by the two algorithmic changes in CLUCB2.

Discussion on martingale bound. An interpretation for the $\sqrt{d}$-improvement comes from comparing (4) and (6). The minimization in (4) has a closed form solution given by $\langle\widehat{\theta}_t, x\rangle - \beta_t\|x\|_{V_t^{-1}}$, with $x := \sum_{i\in S_{t-1}} \phi^i_{a_i}$.[4] While $\langle\widehat{\theta}_t, x\rangle \approx \sum_{i\in S_{t-1}} r^i_{a_i}$ since $\widehat{\theta}_t$ solves the associated regularized least-square problem, $\beta_t\|x\|_{V_t^{-1}} = \widetilde{O}(\sigma\sqrt{d|S_{t-1}|})$, which is larger than the martingale term, which is of order $\widetilde{O}(\sigma\sqrt{|S_{t-1}|})$. The advantage of the martingale argument is that it avoids to explicitly use the linear structure of the reward by building a concentration for the sum of scalar values. As shown above, this allows to derive a bound independent from the dimensionality of the linear parametrization. Nonetheless, notice that in evaluating the quality of the next arm, a minimization over θ is still needed in (6), which brings back the dependency on $\sqrt{d}$ (but on a much smaller term) in the regret analysis, which eventually prevents us from proving an explicit advantage in the final bound. A similar reasoning can be derived for the MAB case (see appendix).

Another interpretation for this $\sqrt{d}$-improvement can be seen when looking at the regret. We start bounding the regret as:

$$R_{\mathrm{CLUCB2}}(n) \leq \sum_{t\in S_n} \left(\mu^t_{a^\star} - \mu^t_{a_t}\right) + |S^b_n|\Delta_h$$

In (Kazerouni et al. 2017), the first term is upper-bounded by the standard regret of LinUCB, while the key step is to bound the regret $|S^b_n|\Delta_h$ incurred while playing the baseline. By exploiting the martingale bound, we can provide a tighter bound for $|S^b_n|$ compared to CLUCB. In (Kazerouni et al. 2017) (after Eq. 19), the authors shows that $\alpha\mu_l|S^b_n| \lesssim 114d^2c_1^2/(\alpha\mu_l)$ where $c_1 = \sigma + \sqrt{\lambda}B$ (ignoring logarithmic terms). By exploiting the martingale bound, we can show that $\alpha\mu_l|S^b_n| \lesssim 10(\sigma + dc_1)/\sqrt{\alpha\mu_l} + 32(\sigma + dc_1)^2/(\alpha\mu_l)$ (see Eq.17 in the appendix of the extended version). This already shows that we have a linear term in d depending only on $1/\sqrt{\alpha}$ and a term quadratic in d as in CLUCB with a much smaller constant. This is a big improvement compared to CLUCB that shows that the martingale indeed provides a $\sqrt{d}$-improvement (and also in $1/\sqrt{\alpha}$). Finally, if we take a very loose upper bound we obtain a term that is smaller by a factor at least 2 compared to the one of CLUCB (formally we obtain that $\alpha\mu_l|S^b_n| \lesssim 48d^2c_1^2/(\alpha\mu_l)$).

Discussion on action selection. The second difference in CLUCB2 is the action selection process. Denote by $\widehat{\mu}_i$ the empirical mean of arm i, let $\mathcal{A}^{\mathrm{UCB}}_t := \arg\max_{i\in[K]}\{\widehat{\mu}_i + \psi^{\mathrm{UCB}}_t(i)\}$ be the set of UCB optimal arms, $\mathcal{A}^+_t := \{i \in [K] : \widehat{\mu}_i + \psi^{\mathrm{UCB}}_t(i) \geq \mu^\star_t\}$ the set of optimistic arms and $\mathcal{C}_t$ the set of arms satisfying the conservative condition. At any time t we can define three events: $E_{1,t} = \{a_t \in \mathcal{A}^{\mathrm{UCB}}_t \wedge a_t \in \mathcal{C}_t\}$, $E_{2,t} = \{a_t \neq b_t \wedge a_t \notin \mathcal{A}^{\mathrm{UCB}}_t \wedge a_t \in \mathcal{C}_t\}$ and $E_{3,t} = \{a_t = b_t\}$. Following the two-step selection process of CLUCB, only $E_{1,t}$ and $E_{3,t}$ can happen. In case $E_{1,t}$, the algorithm behaves like UCB, thus performing exploration that contributes to reduce the regret over time. In case $E_{3,t}$, the regret is equal to $\Delta^t_{b_t}$ and no "progress" is made on the exploration side, but it serves in building conservative budget for later steps. In CLUCB2, event $E_{2,t}$ happens when the UCB arm is not "safe" to play (i.e., $\mathcal{A}^{\mathrm{UCB}}_t \notin \mathcal{C}_t$) but there are other arms that are *safe w.r.t. the baseline*. Interestingly, in this case a_t is indeed *optimistic* w.r.t. the baseline, i.e., $a_t \in \mathcal{C}_t$ and $\widehat{\mu}_i + \psi^{\mathrm{UCB}}_t(i) \geq \mu^t_{b_t}$. In analogy with the OFU principle for $a^\star$, this is a good strategy for performing efficient exploration w.r.t. the baseline policy. One source of improvement comes when a_t is optimistic despite not being the UCB arm (i.e., $a_t \in \mathcal{A}^+_t$ and $a_t \notin \mathcal{A}^{\mathrm{UCB}}_t$). In this case, the time step can be analyzed as in UCB, thus reducing the number of pulls $T_b(n)$ to the baseline arm and its impact to the regret. This is likely to happen in earlier phases of the learning process, where the UCB of all arms tend to be optimistic (i.e., $a_t \in \mathcal{A}^+_t$). Even when a_t is not optimistic w.r.t. $\mu^\star_t$, it may happen that $\mu^t_{a_t} > \mu^t_{b_t}$. In this case, while this step can be still considered as a "conservative", it would contribute for less than $\Delta^t_{b_t}$ regret. Unfortunately, it is difficult to provide theoretical evidence that such events happen often enough to provably reduce the regret. Nonetheless, the fact that the arm is optimistic w.r.t. the baseline (i.e., $\widehat{\mu}_i + \psi^{\mathrm{UCB}}_t(i) \geq \mu^t_{b_t}$) is sufficient to guarantee that the new action selection strategy is never worse than CLUCB.

We can provide an intuition of the impact of the new arm selection through a simple *example*. Consider the situation of $N > 3$ arms, such that $\mu_i > \mu_{i+1}$, for any i. Assume we know the variance of the arms and we use Bernstein inequality in building confidence intervals. Let $\sigma_2 = 0$ and $(\mu_i, \sigma_i)_{i>3}$ such that the probability of being safe and better than arm 2 is negligible. Then the regret can be decomposed by the normal LinUCB term (due to event E_1), the pull of an arm that is optimistic w.r.t. the baseline (i.e., event E_2) and the conservative play. The number of conservative play is thus further reduced due to event E_2, leading to a smaller contribution to the regret. Now, in this specific case, the arm played during event E_2 is w.h.p. always arm 2. Since it is better than the baseline, we have a further improvement to the regret. To conclude, in this example, CLUCB2 will have a regret strictly better than CLUCB.

[3] Notice that there is a typo in Thm.6 in (Kazerouni et al. 2017), as the denominator in the log term of K should be 1.

[4] We remove the contribution of the optimistic arm a_t since it is the same when using martingale or self-normalizing bound.

Checkpoint-based Conservative Exploration

CLUCB2 is designed to get a tighter proxy for (2), but it is still required to be conservative at any time t. This requirement is often too strict in practice, where the conservative condition may be verified only at some "checkpoints" over time. We study the case where the checkpoints are equally spaced every T steps. We still assume that Asm. 3 holds and we redefine the conservative condition such that for some $\alpha \in [0,1]$ and $T \in \mathbb{N}^\star$ a learning agent must satisfy

$$\forall k > 0, \qquad \sum_{t=1}^{kT} \mu_{a_t}^t \geq (1-\alpha) \sum_{t=1}^{kT} \mu_{b_t}^t, \tag{10}$$

which reduces to (2) for $T = 1$. Knowing that the conservative condition is checked every T steps provides the agent with a leeway that can be used to perform more exploration and possibly converge faster towards the optimal policy.

We first derive a conservative condition that can be evaluated at any time $t \in [kT+1, (k+1)T]$ of a phase $k \in \mathbb{N}$ in order to determine whether action a_t is safe. We build this condition such that when selecting an action a_t, we want to ensure that by playing the baseline arm until the next checkpoint (i.e., until $(k+1)T$) , the algorithm would meet the condition (10). Formally, at any step t, we replace (10) with

$$\sum_{i \in S_{t-1}} \mu_{a_i}^i + \sum_{i \in S_{t-1}^b} \mu_{b_i}^i + \mu_{a_t}^t + \alpha((k+1)T - t)\, \mu_l \geq (1-\alpha) \sum_{i=1}^{t} \mu_{b_i}^i, \tag{11}$$

where μ_l is as in Asm. 3, that is to say a lower bound on the average reward of the baseline strategy.

We now modify CLUCB2 to satisfy this constraint. Changing the conservative condition impacts how the algorithm evaluates whether an action is "safe" or not (i.e., if selecting a specific action is compatible with the conservative condition), however the algorithm can still use the bound in (5) to lower bound the sum of the values of actions selected so far, and compute the conservative set at time t as

$$\begin{aligned}\mathcal{C}_t := \Bigg\{ a \in \mathcal{A}_t \setminus \{b_t\} \mid \max\Big\{ \sum_{i \in S_{t-1}} r_{a_i}^i - \psi_L(t), 0 \Big\} \\ + \sum_{i \in S_{t-1}^C} \mu_{b_i}^i + \max\Big\{ \min_{\theta \in \mathcal{C}_t} \left\langle \theta, \phi_a^t \right\rangle, 0 \Big\} \qquad (12) \\ + \alpha \left((k+1)T - t\right) \mu_l \geq (1-\alpha) \sum_{i=1}^{t} \mu_{b_i}^i \Bigg\},\end{aligned}$$

and the arm to pull is obtained by solving the constrained problem (8).

We now proceed by analyzing how this different conservative condition impacts the final regret of CLUCB2, which we rename CLUCB2T to stress the checkpoint-based conservative condition.

Theorem 2. *For any $\delta > 0$, with probability as least $1 - \delta$* CLUCB2T *satisfies condition* (10) *at every checkpoint. Furthermore, let $\tilde{T}_\delta^\alpha := \frac{\alpha\mu_l}{(1-\alpha)\mu_h + \alpha\mu_l} T$, then* CLUCB2T *suffers a regret*

- *If $\tilde{T}_\delta^\alpha \geq C_b(\alpha, \mu_l, \delta)$*

$$\begin{aligned} R(n) \leq O\bigg(\sigma d \log\left(\frac{nD^2}{\lambda d}\right)\sqrt{n} + \frac{\Delta_h}{\alpha\mu_l} \max\Big\{ \mu_h \\ - \frac{\alpha\mu_l}{2}\tilde{T}_\delta^\alpha + d(\sqrt{\lambda}B + \sigma)\sqrt{\tilde{T}_\delta^\alpha \log\left(\frac{\tilde{T}_\delta^\alpha D^2}{\lambda\delta}\right)}, 0 \Big\}, \end{aligned}$$

- *otherwise*

$$\begin{aligned} R(n) \leq O\bigg(\sigma d \log\left(\frac{nD^2}{\lambda d}\right)\sqrt{n} + \frac{\Delta_h d^2}{(\alpha\mu_l)^2} \times \\ \times \left(\sqrt{\lambda}B + \sigma\right)^2 \log\left(\frac{d^2\sqrt{D_0}}{\sqrt{\delta}\alpha\mu_l}\left(\sqrt{\lambda}B + \sigma\right)^2\right)^2 \bigg), \end{aligned}$$

where

$$\begin{aligned} C_b(\alpha, \mu_l, \delta) := 28d^2 \left(\frac{2/3 + \sqrt{\lambda}B + 2\sigma}{\alpha\mu_l}\right)^2 \times \\ \times \ln\left(\frac{696 d^2 D_0}{\delta\left(\alpha\mu_b\right)^2}\left(2/3 + \sqrt{\lambda}B + 2\sigma\right)^2\right)^2, \end{aligned}$$

with $D_0 := \max\{3, 2D^2/\lambda\}$.

This bound illustrate how the length T of each phase may significantly simplify the problem. As T gets larger, satisfying condition (10) becomes easier, the baseline is selected less often and more time is spent in exploring different actions, thus leading to smaller regret. Interestingly, when T is large enough (i.e., $T \geq C_b(\alpha, \mu_l, \delta)$), the conservative contribution has a smaller and smaller impact onto the regret, to the point that the *max* in the second term can become 0, thus reducing the regret to the standard regret bound of LinUCB, with no impact from the conservative constraint.

Alternative checkpoint schemes. While we assumed T to be fixed, it is possible to generalize this result along different lines. If the time between any two checkpoints is known to be lower-bounded by $T_{\min}$, the same analysis could hold by replacing T with $T_{\min}$. Similarly, if T is random from a known distribution, then it is possible to compute the $1 - \delta/2$ quantile of T to recover high-probability guarantees on the conservative properties of the algorithm. Finally, if the checkpoints are completely arbitrary (or even adversarially chosen) then in order to guarantee that (10) is verified at *all* checkpoints, the agent needs to be conservative at every step, thus reducing to condition (2).

Experiments

In this section we provide empirical evidence of the advantage of the Martingale lower-bound and the action selection process in synthetic and real-data problems.

Synthetic Environments

We consider a MAB with $K = 10$ Bernoulli arms whose means are drawn from a uniform distribution, $\mu_i \sim$

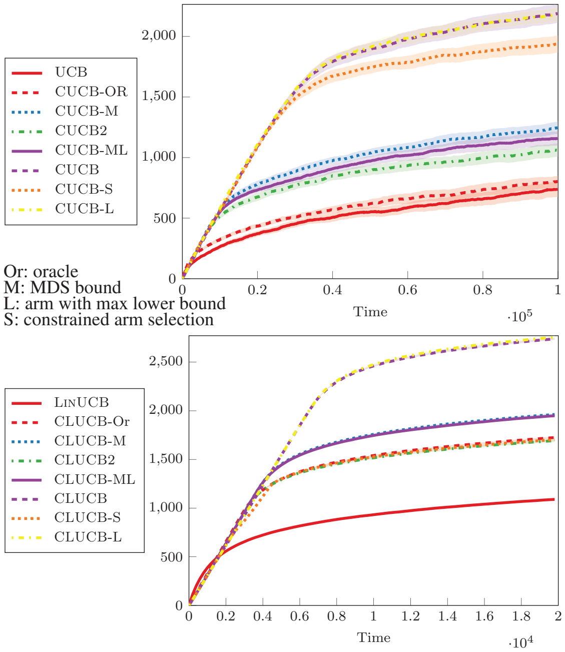

Figure 3: Cumulative regret in synthetic models. *Top:* Bernoulli arms. *Bottom:* linear bandits.

Uniform$([0.25, 0.75])$. The conservative level α is set to 0.05, the horizon n to 10^6 ($T = 1$) and $\delta = 0.01$. We generated 70 different Bernoulli bandit problems (i.e., values of μ_i) and we performed 40 simulations for each. In each problem, we selected the 4th best arm as baseline. Out of the 70 problems, we report the regret curves for the instance where the advantage of CUCB2 w.r.t. CUCB in terms of the average regret at n is the smallest. This provides an estimated worst-case scenario for our comparison (see Appendix for further details and results). We report the performance of UCB and an oracle variant of CUCB (CUCB-Or) where the conservative condition (2) is checked exactly. Furthermore, we test CUCB and a variant of CUCB (CUCB-L) using the action selection process suggested in (Wu et al. 2016), which returns the safe arm with the largest lower bound . Finally, we report an ablation study for CUCB2, where we consider the Martingale lower bound (CUCB-M) and the constrained action selection process (8) (CUCB-S) separately beside the full algorithm (CUCB2). Fig. 3(*top*) shows that the MDS bound alone provides a significant improvement, where the regret is reduced by 43% w.r.t. CUCB's. Interestingly, the action selection process (CUCB-S) is much more effective than CUCB-L and it reduces the regret of CUCB by 12%. Finally, the combination of the two elements (CUCB2) leads to a reduction of the original regret of more than 51%, with a performance which gets much closer to CUCB-Or.

We also evaluated CLUCB2 in the linear setting. We considered a non-contextual case with 30 actions, each defined by a 100-dimensional feature vector. The features and $\theta^\star$ are drawn randomly in the unit ball such that the mean reward of each arm is in $[0, 1]$. The reward noise is drawn from $\mathcal{N}(0, 0.1^2)$ and the baseline arm is the 6th best action. We set $\lambda = 0.5$, $\delta = 0.01$ and $\alpha = 0.05$. We generated 70 models and for each model we averaged the results over 40 runs. As in the MAB case, we report the results for the model with the smallest advantage for CLUCB2 w.r.t. CLUCB (Fig. 3(*bottom*)). Contrary to the MAB setting, the main improvement is obtained by CLUCB-S, whose performance matches CLUCB2 and even the oracle variant, corresponding to an improvement of 38% compared to CLUCB. As reported in the appendix in the average case (over models), we observe similar behaviors and performance improvements as in the MAB setting. Finally, to provide an idea of how many times event E_2 occurs, that is to say the number of times where the algorithm does not play the optimistic action and not the arm suggested by the baseline strategy, we performed tests in synthetic linear setting with baseline being the third-best arm. On average over multiple models, the percentage of time event E_2 happened in the first 5000 steps (25% of overall time) is 38.7% (±9%). This shows that potentially we have played something better than baseline and for sure we have gained information (in contrast to playing the baseline).

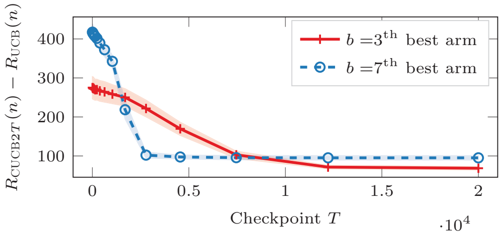

Figure 4: Relative performance between CUCB2T and UCB in synthetic MAB setting.

Checkpoint-based Condition. We compare the effect of the checkpoint T on the regret of the algorithms. We report the results for Bernoulli arms, the linear experiments can be found in Appendix. In this case, the horizon is set to $n = 20000$, all the other parameters are unchanged. We generated 15 (integer) checkpoint values logarithmic space between 1 and n. Fig. 4 shows the difference in the regret between CUCB2T and UCB as a function of T. As expected, the difference decreases as T increases since the condition becomes less strong. Note that even for $T = n$, CUCB2T and UCB are different since CUCB2T might discard UCB optimal arms in order to be safe. In order to have the same behavior, T should be put sufficiently large in order to overcome the pessimistic estimate used in the condition. Finally, the improvement provided by the new condition is proportional to the quality of the baseline. The stronger the baseline, the less is the margin for performing better exploration than playing the baseline itself.

Dataset-based Environments

Fig. 5 reports the results using the Jester Dataset (Goldberg et al. 2001) that consists of joke ratings in a continuous scale from -10 to 10 for 100 jokes from a total of

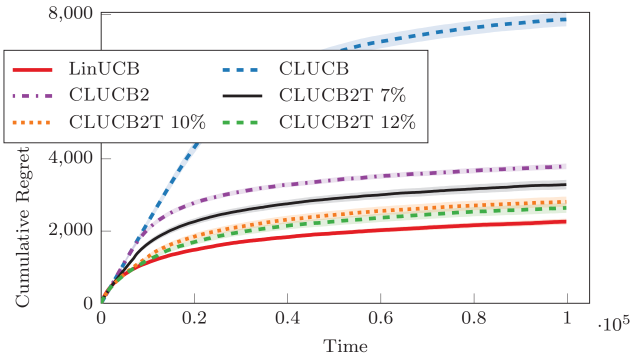

Figure 5: Average regret over multiple users of the Jester dataset

73421 users. We consider the cold start problem: a new user arrives and we need to learn her preferences (i.e., $\theta^\star$). We use the features extracted via a low-rank matrix factorization ($d = 35$) to represents the actions (i.e., the jokes). We consider a complete subset of 40 jokes and 19181 users rating all the 40 jokes. The preference of the new user is randomly selected from the 19181 users and mean rewards are normalized in $[0, 1]$. The reward noise is $\mathcal{N}(0, 0.1^2)$, the horizon is $T = 10^5$, $\alpha = 0.01$, $\delta = 0.01$ and $\lambda = 0.5$ (see appendix). We report the results averaged over 100 randomly selected users and for each user we performed 5 runs. The baseline is the 10^{th} best arm. We also report the regret of CLUCB2T with a checkpoint horizon T equal to $5\%, 10\%$ or 12% of the horizon n. This experiment confirms that CLUCB2 performs best, with a regret that is less than half of CLUCB. Furthermore, the results confirm that as the checkpoints become sparser, the performance of CLUCB2 approaches the one of LinUCB.

Conclusion

We introduced CLUCB2, a novel conservative exploration algorithm for linear bandit that matches existing regret bound and outperforms state-of-the-art algorithms in a number of empirical tests. In this paper, we also proposed a first direction to relax the conservative condition towards a more realistic scenario. Important directions for future work are: identify alternative conservative exploration constraints that are directly motivated by specific applications, extend the current algorithms beyond linear bandit towards the more challenging reinforcement learning setting.

References

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

Auer, P.; Cesa-Bianchi, N.; and Fischer, P. 2002a. Finite-time analysis of the multiarmed bandit problem. *Machine Learning Journal* 47:235–256.

Auer, P.; Cesa-Bianchi, N.; and Fischer, P. 2002b. Finite-time analysis of the multiarmed bandit problem. *Machine learning* 47(2-3):235–256.

Bottou, L.; Peters, J.; Quinonero-Candela, J.; Charles, D.; Chickering, D.; Portugaly, E.; Ray, D.; Simard, P.; and Snelson, E. 2013. Counterfactual reasoning and learning systems: The example of computational advertising. *Journal of Machine Learning Research* 14:3207–3260.

Freedman, D. A. 1975. On tail probabilities for martingales. *the Annals of Probability* 3(1):100–118.

Goldberg, K.; Roeder, T.; Gupta, D.; and Perkins, C. 2001. Eigentaste: A constant time collaborative filtering algorithm. *information retrieval* 4(2):133–151.

Katariya, S.; Kveton, B.; Wen, Z.; and Potluru, V. K. 2019. Conservative exploration using interleaving. In *AISTATS*, volume 89 of *Proceedings of Machine Learning Research*, 954–963. PMLR.

Kazerouni, A.; Ghavamzadeh, M.; Abbasi, Y.; and Roy, B. V. 2017. Conservative contextual linear bandits. In *NIPS*, 3910–3919.

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.

Mansour, Y.; Slivkins, A.; and Syrgkanis, V. 2015. Bayesian incentive-compatible bandit exploration. In *Proceedings of the Sixteenth ACM Conference on Economics and Computation*, 565–582. ACM.

Petrik, M.; Ghavamzadeh, M.; and Chow, Y. 2016. Safe policy improvement by minimizing robust baseline regret. In *Advances in Neural Information Processing Systems*, 2298–2306.

Swaminathan, A., and Joachims, T. 2015. Counterfactual risk minimization: Learning from logged bandit feedback. In *Proceedings of The 32nd International Conference on Machine Learning*.

Thomas, P.; Theocharous, G.; and Ghavamzadeh, M. 2015a. High confidence off-policy evaluation. In *Proceedings of the Twenty-Ninth Conference on Artificial Intelligence*.

Thomas, P.; Theocharous, G.; and Ghavamzadeh, M. 2015b. High confidence policy improvement. In *Proceedings of the Thirty-Second International Conference on Machine Learning*, 2380–2388.

Wu, Y.; Shariff, R.; Lattimore, T.; and Szepesvári, C. 2016. Conservative bandits. In *ICML*, volume 48 of *JMLR Workshop and Conference Proceedings*, 1254–1262. JMLR.org.